

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering



Project Report on

**Dynamic Queuing Algorithms for Optimized
Healthcare Appointment and Patient Flow
Management in OPD Systems**

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2024-25

Submitted by

Sarvesh Dongare [D17A-14]
Aniket Pradhan [D17A-47]
M.Kaif Qureshi [D17A-48]
Parth Wande [D17A-68]

Project Mentor

Dr. Nupur Giri
(2024-25)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering



Certificate

This is to certify that **M.Kaif Qureshi, Aniket Pradhan ,Parth Wande, Sarvesh Dongare** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "**Dynamic Queuing Algorithms for Optimized Healthcare Appointment and Patient Flow Management in OPD Systems**" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor **Dr. Nupur Giri** in the year 2024-25 .

This thesis/dissertation/project report entitled **Dynamic Queuing Algorithms for Optimized Healthcare Appointment and Patient Flow Management in OPD Systems** by **M.Kaif Qureshi, Aniket Pradhan ,Parth Wande, Sarvesh Dongare** is approved for the degree of **B.E. Computer Engineering**.

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide:

Project Report Approval

For

B. E (Computer Engineering)

This thesis/dissertation/project report entitled ***Dynamic Queuing Algorithms for Optimized Healthcare Appointment and Patient Flow Management in OPD Systems*** by ***M.Kaif Qureshi, Aniket Pradhan, Parth Wande, Sarvesh Dongare*** is approved for the degree of **B.E. Computer Engineering.**

Internal Examiner

External Examiner

Head of the Department

Principal

Date:

Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

M.Kaif Qureshi (48)

Aniket Pradhan (47)

Parth Wande (68)

Sarvesh Dongare (14)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Dr. Nupur Giri (Project Guide)** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair**, for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department
COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop a professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

Chapter No.	Title	Page No.
1	Introduction	11
	1.1 Introduction to the project	11
	1.2 Motivation for the project	11
	1.3 Problem Definition	11
	1.4 Existing Systems	12
	1.5 Lacuna of the Existing Systems	12
	1.6 Relevance of the Project	13
2	Literature Survey	14
	A Brief overview of Literature Survey	14
	2.1 Research Papers <ul style="list-style-type: none"> a. Abstract of the research paper b. Inference drawn from the paper 	14
	2.2 Unified Inference from Literature	16
	2.3 Comparison with the Existing Systems:	17
3	Requirement Gathering for the proposed System	18
	3.1 Introduction to Requirement Gathering	18
	3.2 Functional Requirements	18
	3.3 Non-Functional Requirements	19
	3.4 Hardware, Software, Technology and tools utilised	19
	3.5 Constraints	20
4	Proposed Design	21
	4.1 Block diagram of the system	21
	4.2 Modular design of the system	22
	4.3 Detailed Design (Flowchart)	22
	4.4 Methodology/Block Diagram	25
	4.5 Project Scheduling & Tracking using Timeline / Gantt Chart	25
5	Implementation of the Proposed System	24

	5.1	Methodology employed for development	26
	5.2	Algorithms and flowcharts for the respective modules developed	26
	5.3	Datasets source and utilisation	27
6		Testing of the Proposed System	26
	6.1	Introduction to testing	28
	6.2	Types of tests Considered	28
	6.3	Various test case scenarios considered	29
	6.4	Inference drawn from the test cases	30
7		Results and Discussions	31
	7.1	Screenshots of User Interface (UI) for the respective module	31
	7.2	Graphical Interpretations	33
8		Conclusion	34
	8.1	Limitations	34
	8.2	Conclusion	34
	8.3	Future Scope	35
		References	36
		Appendix	37
	a	Paper I	37
	b	Plagiarism Report of Paper I	44
	c	Project review sheet	45

List of Figures:

Fig no.	Heading	Page no.
4.1	Block Diagram	22
4.2	FlowChart Diagram	23
4.3.1	Data flow diagram 0 level	23
4.3.2	Data flow diagram 1 level	24
4.3.3	Data flow diagram 2 level	24
4.4	Methodology Diagram	25
4.5	Gantt Chart	25
7.1.1	Screenshot for Web dashboard	31
7.1.2	Screenshot for Web health card	31
7.1.3	Screenshot for Token Dashboard	32
7.1.4	Screenshot for Bed Availability Dashboard	32
7.1.5	Screenshot for OPD Dashboard	32
7.1.6	Screenshot for Mobile UI	33
7.2.1	PDF of Waiting Times	33
7.2.2	CDF of Waiting time	33

List of Tables:

Table no.	Heading	Page no.
2.1	Literature Survey	14
2.3	Comparison Table of Existing systems	17
3.4	Hardware and Software Requirements	20
6.3.1	Table of processing times	29
6.3.2	Table Comparing The proposed and traditional systems	29

Abstract

Efficient management of Outpatient Departments (OPDs) is a cornerstone of delivering timely and high-quality healthcare services. As patient volumes continue to rise and medical resources remain constrained, traditional appointment scheduling systems often struggle to keep up. Long wait times, overcrowded waiting rooms, and missed appointments are common challenges faced by both patients and healthcare providers. To address these issues, we introduce a dynamic queuing algorithm that brings intelligence and adaptability to the forefront of OPD operations.

Our proposed solution is designed to optimize appointment scheduling and streamline patient flow by factoring in real-time variables such as patient priority, type of appointment, urgency levels, and current resource availability (e.g., doctors, nurses, diagnostic equipment, and consultation rooms). Unlike static systems, which operate on a first-come, first-served or time-slot basis, this algorithm dynamically adjusts patient queues as conditions change—ensuring that those who need urgent care are prioritized while maintaining an efficient flow for all other patients.

A defining innovation of our approach is its real-time adaptability. When unforeseen events occur—such as emergency walk-ins, doctor unavailability, or sudden surges in patient volume—the algorithm recalibrates appointment slots and reallocates healthcare resources instantly. This flexibility not only reduces bottlenecks but also significantly improves appointment adherence rates and reduces the number of patients who leave without being seen.

To validate our system, we tested the algorithm in a simulated hospital environment that mirrors the complexities of real-world healthcare settings. The results were promising: average wait times were reduced, resource utilization improved, and patient satisfaction metrics increased. These outcomes demonstrate the potential of dynamic scheduling to transform how OPDs are managed—making them more responsive, efficient, and patient-centric.

In an era where digital transformation is reshaping every aspect of healthcare, our dynamic queuing system represents a significant step toward building smarter, more resilient healthcare infrastructures. By aligning technological innovation with patient care goals, hospitals can not only enhance operational efficiency but also ensure that patients receive the timely attention they deserve.

Chapter 1: Introduction

1.1. Introduction:

India's healthcare system is making strides to improve access and quality of care, but it still faces significant operational challenges. In many hospitals, outdated or manual systems for managing OPD queues result in long wait times, especially for patients in urgent need, creating frustration and inefficiency. Managing bed availability is another area where delays in admissions and discharges put a strain on resources and impact patient outcomes. The patient admission process itself is often manual and fragmented, leading to unnecessary delays and administrative headaches. On top of this, inventory management struggles with real-time tracking, causing shortages or wastage of essential medicines and supplies. Finally, hospitals often operate in isolation from broader city-wide health systems, making it harder to share patient data, coordinate care, and respond effectively to emergencies. These gaps not only slow down hospital operations but also compromise the overall patient experience.

1.2. Motivation:

The motivation behind "IndiHealth" stems from the critical need to enhance healthcare delivery by addressing operational inefficiencies through technology. The project targets several key challenges: reducing outpatient wait times through dynamic prioritization systems; improving real-time bed management to optimize admissions and discharges; streamlining patient registration processes; implementing automated inventory tracking to prevent medicine shortages; and connecting hospital systems with city-wide health networks for better coordination.

To achieve these improvements, IndiHealth will leverage advanced technologies including blockchain for secure data management, telemedicine capabilities for remote care delivery, and artificial intelligence for predictive analytics. This comprehensive approach aims to transform both patient experience and resource utilization, creating a more efficient, responsive healthcare ecosystem that benefits patients, healthcare providers, and the broader community through streamlined operations and enhanced care delivery.

1.3. Problem Definition:

The healthcare system in India faces several operational challenges that hinder the efficient delivery of services and compromise patient care. Hospitals struggle with outdated processes in outpatient department (OPD) queuing, bed management, patient admissions, and inventory control, leading to delays and inefficiencies. These issues not only prolong patient wait times but also place undue pressure on healthcare staff, negatively impacting the overall patient experience.

Additionally, hospitals often operate in isolation from broader city-wide health systems, which limits their ability to share patient data, coordinate care, and respond effectively to emergencies.

These challenges underscore the need for a comprehensive solution that leverages modern technologies to optimize hospital operations, streamline patient flow, and improve overall healthcare management and patient outcomes.

1.4. Existing Systems:

Current hospital management systems are often outdated, fragmented, and inefficient, impacting patient care and operational effectiveness. Many hospitals still rely on manual processes, lack real-time updates, and do not integrate with broader healthcare networks, creating significant challenges across multiple operational areas.

Queuing systems in most hospitals use manual or static digital approaches that fail to account for patient urgency. This one-size-fits-all approach creates unnecessary bottlenecks and extends wait times, particularly for patients requiring prompt attention. The inability to dynamically adjust queues based on medical priority compromises both efficiency and quality of care.

Bed availability management suffers from limited real-time tracking capabilities. Hospital staff often rely on manual updates for bed status, resulting in inefficient assignments and delayed admissions or discharges. This outdated approach to resource management creates unnecessary delays in patient care transitions and reduces overall capacity utilization.

Patient admission processes remain largely fragmented and manual in many healthcare facilities. These workflows typically involve redundant paperwork and disjointed data flow between departments, creating administrative burdens for staff and frustrating experiences for patients. The lack of streamlined digital solutions leads to preventable delays and increased potential for errors.

Inventory management systems in traditional hospital settings frequently fail to provide real-time tracking of medical supplies and equipment. The reliance on manual restocking checks increases the risk of critical stockouts or unnecessary overstocking. Without automated alerts and predictive analytics, hospitals struggle to maintain optimal inventory levels to support patient care.

Perhaps most concerning is the lack of city-wide integration in hospital management systems. Most operate as isolated entities that do not connect with broader healthcare networks or emergency response services. This limitation severely restricts the potential for coordinated care during mass casualty events or public health emergencies when resource sharing becomes critical.

IndiHealth aims to address these inefficiencies through an integrated approach featuring real-time updates, dynamic queuing based on patient needs, automated processes for administrative tasks, and seamless integration with city-wide healthcare systems.

1.5. Lacuna of the Existing System:

While existing hospital management systems provide basic operational support, they fall short in several key areas, leaving significant gaps that impact healthcare efficiency and patient care. These shortcomings include a lack of dynamic queuing systems that can prioritize patients based on urgency in real-time, inefficient bed management without centralized visibility into availability, and persistence of manual or semi-automated processes for patient admissions and other critical operations.

Additionally, traditional systems fail to offer adequate inventory control with real-time tracking or automated restocking alerts, resulting in stockouts and wastage of critical medical supplies. Perhaps most critically, most existing hospital systems operate in isolation, lacking integration with broader city-wide healthcare networks or emergency services, which hampers coordinated care delivery during routine operations and especially during emergencies or mass casualty events. These limitations collectively impede healthcare facilities from delivering efficient, responsive, and patient-centered care.

1.6. Relevance of the Project:

In today's digital era, where education is rapidly shifting online, the need for a secure and transparent examination system has become more critical than ever. With repeated instances of paper leaks, result manipulation, and unauthorized access in traditional and online examination systems, there is a growing demand for a more trustworthy solution.

This project addresses these concerns by introducing a decentralized examination portal built on Web3 technologies. Leveraging blockchain ensures data immutability and transparency, while IPFS provides decentralized and secure storage of exam materials. Smart contracts enable automated and unbiased grading, significantly reducing human errors and intervention. AES encryption ensures that sensitive information remains protected throughout the exam lifecycle.

By integrating these modern technologies, the project aligns with current educational needs and offers a future-ready solution that enhances fairness, accountability, and data integrity. It holds immense relevance for institutions seeking to upgrade their assessment processes and restore trust among students and stakeholders.

Chapter 2: Literature Survey

A. Overview of Literature Survey:

In recent years, the implementation of dynamic queuing models in healthcare systems has gained significant traction as a strategy to enhance patient flow and reduce wait times. Traditional queuing approaches often struggle to adapt to the variability in patient arrivals and service needs, which can lead to inefficiencies in emergency departments and outpatient settings.

2.1. Research Papers:

sr no.	Papers			
1	Title:	Application of Queuing Theory to Optimize Waiting-Time in Hospital Operations		
	Authors:	D. Yaduvanshi, A. Sharma, P. V. More	Year	2019
	Objective:	Optimize waiting times in hospitals using queuing theory		
	Methodology:	Queuing Theory		
	Key Findings:	Reduced waiting times for hospital operations		
	Relevance to Project:	Relevant for improving patient flow and reducing waiting times in a hospital management system		
2	Title:	Improving Outpatient Flow at an Indian Ophthalmic Hospital		
	Authors:	Y. Daulatani, S. Kumar, O. Vaidya	Year	2016
	Objective:	Improve outpatient flow at an ophthalmic hospital		
	Methodology:	Process flow analysis, simulation		
	Key Findings:	Improved outpatient flow and patient management		
	Relevance to Project:	Useful for understanding outpatient flow management in hospital systems		
3	Title:	Assessing and controlling the impact of hospital capacity planning on the waiting time		
	Authors:	N. Dellaert, E. Cayiroglu, J. Jeunet	Year	2015
	Objective:	Assess impact of capacity planning on hospital waiting times		
	Methodology:	Hospital capacity planning, queuing models		
	Key Findings:	Hospital capacity impacts waiting times, better management improves patient care		
	Relevance to Project:	Relevant for managing patient capacity and waiting times in healthcare facilities		

4	Title:	The effect of inventory management on firm performance		
	Authors:	D. P. Koumanakos	Year	2008
	Objective:	Examine the effect of inventory management on firm performance		
	Methodology:	Inventory management models		
	Key Findings:	Effective inventory management positively impacts firm		
	Relevance to Project:	Important for managing inventory in hospitals, especially for medical supplies		
5	Title:	Inventory control: its principles and application		
	Authors:	A. Singh, S. K. Rasania, K. Barua	Year	2022
	Objective:	Examine the effect of inventory management on firm performance		
	Methodology:	Inventory management models		
	Key Findings:	Effective inventory management positively impacts firm		
	Relevance to Project:	Important for managing inventory in hospitals, especially for medical supplies		
6	Title:	Classification of medicines and materials in hospital inventory management: a multi-criteria analysis		
	Authors:	A. G. de Assis, A. F. A. dos Santos, L. A. dos Santos	Year	2022
	Objective:	Classify medicines and materials for effective hospital inventory management		
	Methodology:	Multi-criteria analysis		
	Key Findings:	Multi-criteria methods help optimize inventory classification and management		
	Relevance to Project:	Useful for classifying and managing hospital inventory in a structured manner		
7	Title:	Patients' and healthcare providers' perceptions of a mobile portal application for hospitalized patients		
	Authors:	K. J. O'Leary, R. K. Sharma, A. Killarney, L. S. O'Hara, M. E. Lohman, E. Culver, D. M. Liebovitz, K. A. Cameron	Year	2016
	Objective:	Examine perceptions of patients and healthcare providers on a mobile portal app for hospitalized patients		
	Methodology:	Survey of perceptions, mobile app evaluation		

	Key Findings:	Positive feedback from both patients and providers on the use of mobile apps in hospitals		
	Relevance to Project:	Can help in designing mobile solutions for hospital management and patient engagement		
8	Title:	Growth in the care of older patients by hospitalists in the United States		
	Authors:	Y. F. Kuo, G. Sharma, J. L. Freeman, J. S. Goodwin	Year	2009
	Objective:	Analyze the growth of hospitalist care for older patients in the U.S.		
	Methodology:	Statistical analysis, healthcare data review		
	Key Findings:	Increasing trend of hospitalist care for older patients, improving outcomes		
	Relevance to Project:	Relevant for analyzing healthcare delivery models, especially for elderly patients		
9	Title:	Ability of hospitalized patients to identify their in-hospital physicians		
	Authors:	V. Arora, S. Gangireddy, A. Mehrotra, R. Ginde, M. Tormey, D. Meltzer	Year	2009
	Objective:	Study whether hospitalized patients can identify their in-hospital physicians		
	Methodology:	Surveys, patient interviews		
	Key Findings:	Many hospitalized patients cannot correctly identify their physicians		
	Relevance to Project:	Relevant for improving communication and identification systems in hospital management		

2.2 Literature Survey

2.2. Unified Inference from Literature:

The synthesis of research on hospital management optimization reveals a transformative opportunity to address longstanding healthcare inefficiencies through integrated technological solutions. Multiple studies confirm that queuing theory application significantly reduces patient waiting times, while strategic capacity planning and structured inventory management approaches yield measurable improvements in resource utilization and organizational performance. The documented receptiveness to technological interventions among stakeholders, despite persistent communication gaps between providers and patients, validates the need for comprehensive systems like IndiHealth that simultaneously address patient flow, resource allocation, inventory control, and provider-patient communication. By leveraging these interconnected insights, healthcare facilities can implement data-driven solutions that enhance operational efficiency, improve care coordination, and ultimately transform the patient experience, though successful implementation will require careful consideration of integration challenges, user adoption strategies, and compliance with healthcare regulations.

2.3. Comparison with the Existing Systems:

Feature	Existing Systems	Proposed System
Queuing System	Operates on a static first-come, first-serve basis with no dynamic adjustment for patient urgency, leading to inefficiencies and long wait times.	Dynamic queuing system that prioritizes patients based on urgency, provides real-time updates, and improves overall patient flow management.
Bed Availability Management	Offers bed tracking but lacks automated real-time allocation, leading to delays in admissions and inefficient bed management.	Real-time bed tracking with automated bed allocation based on patient condition, ensuring optimal resource use and faster patient admissions.
Patient Admission Process	Manual processes for patient admission are still common, with redundant paperwork and fragmented data flow between departments.	Automated and integrated admission processes with online pre-admission forms, reducing paperwork and streamlining patient data across systems.
Inventory Management	Limited real-time inventory tracking with manual restocking checks, leading to shortages and wastage.	Real-time inventory tracking with automated restocking alerts, ensuring timely replenishment and reducing wastage or stockouts.
City-Wide Integration	Focuses primarily on outpatient scheduling and does not integrate with broader city-wide or emergency response systems.	Seamless integration with city-wide health systems and emergency response units, improving coordination and response times in critical scenarios.
Patient Flow Analytics	Provides basic reports but lacks predictive analytics for patient flow and staff workload, leading to inefficiencies during peak times.	Predictive analytics to forecast patient flow and manage workloads, optimizing resource allocation and reducing hospital bottlenecks.

2.3 Comparison Table of Existing systems

Chapter 3: Requirement Gathering for the Proposed System

3.1 Introduction to Requirement Gathering

Requirement gathering is a critical phase in the development of any system, ensuring that the final product aligns with user needs and expectations. In the context of our blockchain-based online examination system, this process involves identifying and documenting the functional and non-functional requirements, as well as specifying the hardware and software resources necessary for implementation.

The requirement gathering process encompasses the following steps:

- **Identifying Stakeholders:** Recognizing all parties involved or impacted by the system, including students, educators, administrators, and technical support teams.
- **Establishing Objectives:** Defining the goals the system aims to achieve, such as enhancing security, ensuring data integrity, and providing a user-friendly interface.
- **Eliciting Requirements:** Collecting detailed requirements from stakeholders through surveys, interviews, and observations.
- **Documenting Requirements:** Clearly recording the gathered requirements for reference and validation.
- **Confirming Requirements:** Reviewing the documented requirements with stakeholders to ensure accuracy and completeness.
- **Prioritizing Requirements:** Determining the importance of each requirement to focus development efforts effectively.

3.2 Functional Requirements

Functional requirements define the specific behaviors and functionalities of the system. For our Dynamic Queuing Algorithms for Optimized Healthcare Appointment and Patient Flow Management in OPD Systems, the key functional requirements are:

- **Mobile Application:** The system shall provide patients with a mobile app that allows them to Book OPD (Outpatient Department) appointments. Locate nearby hospitals based on their location using integrated maps. Request an ambulance during emergencies.
- **Hospital Web Dashboard:** The hospital staff shall have access to a web dashboard that displays Real-time data on queue management, including token generation and estimated wait times for patients. Bed availability and patient admission/discharge status in real-time. Analytics on employee schedules, hospital equipment usage, and inventory levels. Reports on hospital operations, such as patient demographics, bed occupancy, and equipment usage trends.
- **Resource Allocation Insights:** The system shall provide insights into resource allocation by showing correlations between medical staff, equipment, and patient needs. The system will highlight how different combinations of resources (e.g., ICU beds with specific types of equipment) affect patient care, recovery times, and treatment success rates.
- **Patient and Resource Compatibility Analysis:** The system shall analyze the compatibility between a patient's medical needs and the hospital's available resources. For example, critical patients requiring ICU beds will be matched with the closest available ICU with the necessary medical equipment and staff expertise.

- **Emergency Response Coordination:** The system shall support emergency response coordination by integrating with city-wide emergency services. It will manage real-time ambulance tracking, hospital allocations, and automatic alerts to ensure efficient emergency handling.
- **Inventory and Equipment Management:** The system shall track hospital inventory, including medicines, consumables, and medical equipment. It will generate automated alerts for low stock, schedule restocking, and report on inventory usage trends.
- **Patient History and Data Integration:** The system shall maintain an integrated record of patient medical history, allowing doctors and healthcare providers to access relevant data quickly. This includes past appointments, lab tests, treatment history, and future appointments for seamless care continuity.
- **Reporting and Analytics:** The system shall generate detailed reports, including Bed occupancy statistics, Inventory usage reports, Patient flow and demographics analytics, Employee performance metrics.
- **Lab and Pharmacy Integration:** The system shall coordinate patient lab tests and prescriptions with the hospital's pharmacy. Critical patients will be prioritized for lab tests and medication collection, reducing wait times for urgent care.
- **Follow-up and Appointment Scheduling:** Patients shall be able to arrange follow-up appointments through the mobile app or web portal after their treatment. The system will send reminders and maintain a record of scheduled follow-ups for both patients and healthcare workers.

3.3 Non-Functional Requirements

Non-functional requirements specify the quality attributes the system must possess. For our proposed system, these include:

- **Performance:** The system shall handle data processing in real time to ensure updates on bed availability, queue status, and ambulance tracking are instantly visible. It shall efficiently manage concurrent user requests during peak hours, ensuring minimal delays in data retrieval and user interaction.
- **Scalability:** The system shall be designed to scale as the hospital grows, allowing the addition of new departments, staff, and resources without requiring system overhauls. It shall support multiple hospitals within a network, handling data flow between them while maintaining performance consistency.
- **Reliability:** The system shall maintain high availability with an uptime target of 99.5%. It shall include mechanisms for error handling, ensuring smooth recovery from any failures and providing meaningful feedback to users when issues arise.
- **Security and Compliance:** The system shall implement strong authentication, authorization, and encryption to protect sensitive patient data. Compliance with healthcare data privacy regulations such as HIPAA, GDPR, and NHS standards shall be ensured.
- **Usability:** The system interfaces (mobile app, web dashboard) shall be user-friendly, requiring minimal effort for users to navigate and access functionalities. It shall provide clear instructions and an intuitive flow, especially for non-technical users like patients and some healthcare workers.
- **Data Quality:** The system shall regularly validate and clean data to ensure accuracy and prevent duplicates, ensuring the integrity of patient records, inventory counts, and hospital resource management.

3.4 Hardware, Software, Technology, and Tools Utilized

Development Tools:

Category	Technologies/Tools
Frontend and UI	ReactJS (Website), React Native (Mobile Application), Figma
Backend/Database	Python, Flask, Express, NodeJS, MongoDB, SQL
Data Visualization	Jupyter Notebook, D3.js
Cloud Service	Amazon Web Service (AWS)

Table 3.4 Development tools for indiHealth

3.5 Constraints

Data Source Limitations: The system depends on reliable data from external sources (e.g., ambulance services, external lab test providers) for real-time updates. Any changes in these external systems or their APIs may impact system functionality.

Time Constraints: The project must be completed within a specified timeline, which may limit the number of features included in the initial release. Regular updates will follow to expand functionality.

Chapter 4: Proposed Design

Our solution - a mobile app for patients and a web application for hospitals—aims to optimize patient wait times using a dynamic greedy algorithm and smart token generation. Additionally, it features a centralized pharmacy and inventory management system with a focus on city-wide integration. Our System uses Greedy Dynamic Approach to prioritize and Speed up the queuing process. Our System stores the patient's previous health records using blockchain protected via KYC. Proposed System will have a Website and a Mobile App functioning as our Frontend. System will make an Emergency Response System for city wide emergencies. Display the nearest and optimized route for hospitals and clinics near the users.

4.1 Block Diagram of the proposed system:

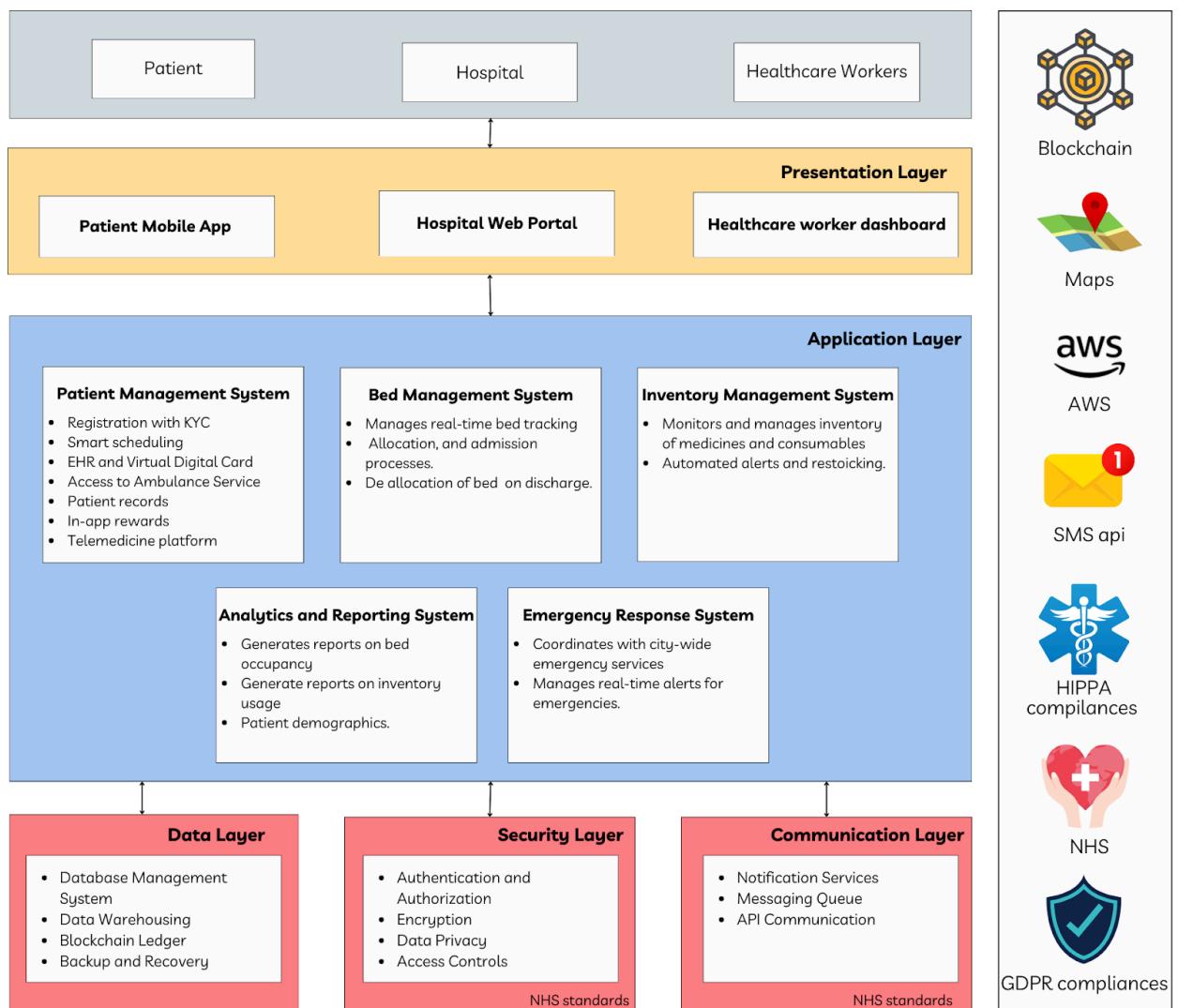


Fig 4.1: Block Diagram/System Design

This block diagram provides an overview of the **IndiHealth System Architecture**, showing various layers and components essential for managing patient and hospital operations.

1. User Layer: Patient, Hospital, and Healthcare Workers: The system users interact through different interfaces depending on their role.

2. Presentation Layer:

- **Patient Mobile App:** A mobile app for patients to access services such as registration, appointment booking, and medical history.
- **Hospital Web Portal:** A web interface used by hospital staff to manage hospital operations like bed allocation and patient records.
- **Healthcare Worker Dashboard:** A specialized interface for healthcare workers, showing vital stats, task management, and more.

3. Application Layer:

This is the core of the system where various modules work together to provide functionalities:

- **Patient Management System:** This module manages patient information such as KYC, scheduling, Electronic Health Records (EHR), and virtual cards. It also offers access to ambulance services and telemedicine features.
- **Bed Management System:** Tracks bed availability and handles allocations, admissions, and deallocation upon patient discharge.
- **Inventory Management System:** Monitors the inventory of medicines and consumables, triggering automated alerts for restocking.
- **Analytics and Reporting System:** Generates reports on bed occupancy, inventory usage, and patient demographics.
- **Emergency Response System:** Coordinates with city-wide emergency services and generates real-time alerts for emergency management.

4. Data Layer: Handles database management, data warehousing, blockchain ledger storage, and backup/recovery mechanisms.

5. Security Layer: Ensures system security with authentication, authorization, encryption, and access controls to maintain data privacy and comply with NHS standards.

6. Communication Layer: Manages notification services, messaging queues, and API communications between system components.

7. External Integrations:

The system is integrated with various external services for added functionalities:

- **Maps:** For location-based services.
- **AWS:** Cloud infrastructure for hosting and scaling services.
- **SMS API:** For communication with patients and healthcare workers.
- **Compliance Standards:** HIPAA, GDPR, and NHS compliance for secure handling of sensitive health data.

4.2. Flowchart for the proposed system:

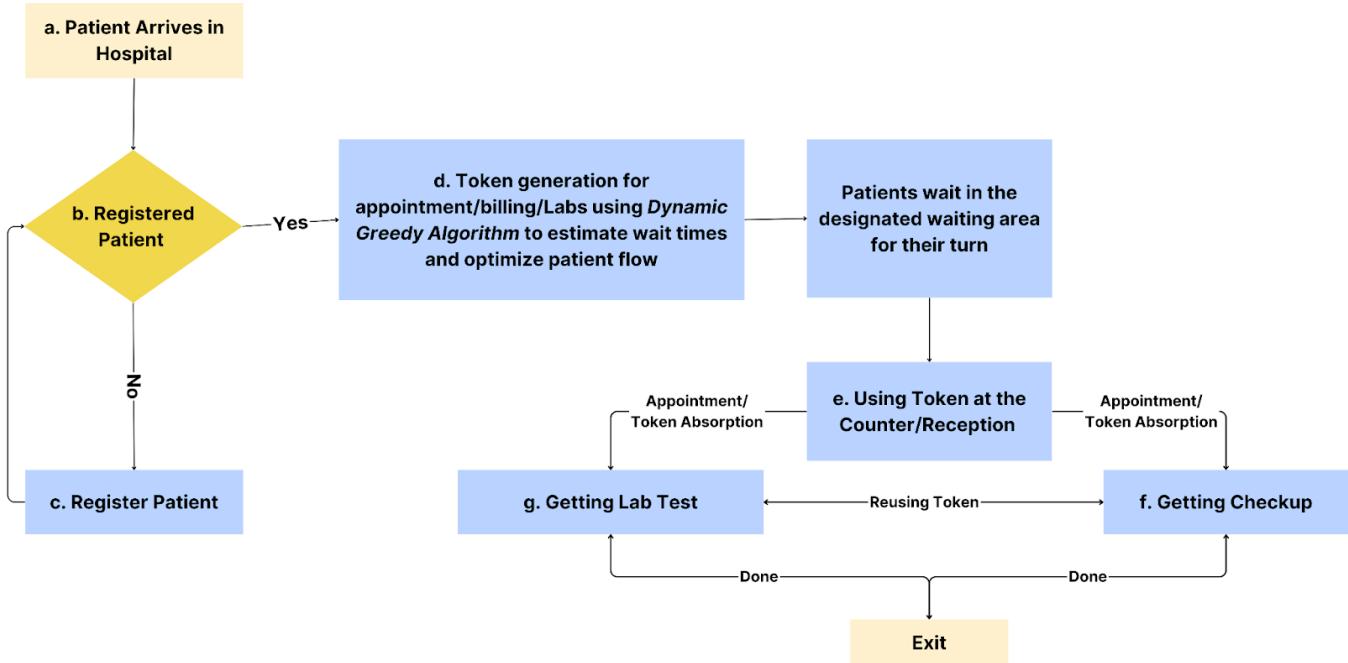


Fig 4.2: Flowchart for the proposed system

After registration (or for previously registered patients), the system generates a unique token that serves as a digital identifier across various hospital services including appointments with doctors, billing and payment at the counter, and lab tests and diagnostics (Step d). The token system assigns each patient to an optimized queue based on their specific service needs. The dynamic queue management algorithm provides patients with continuously updated wait time estimates that consider the **current number of patients in the queue, historical and real-time average service times, available resources** (doctors, lab equipment, counters), and **patient-specific factors** (priority, condition severity). Patients are directed to appropriate waiting areas with real-time updates about their position in the queue delivered via mobile notifications or display screens.

4.3. Data Flow diagram(Level 0,1,2):

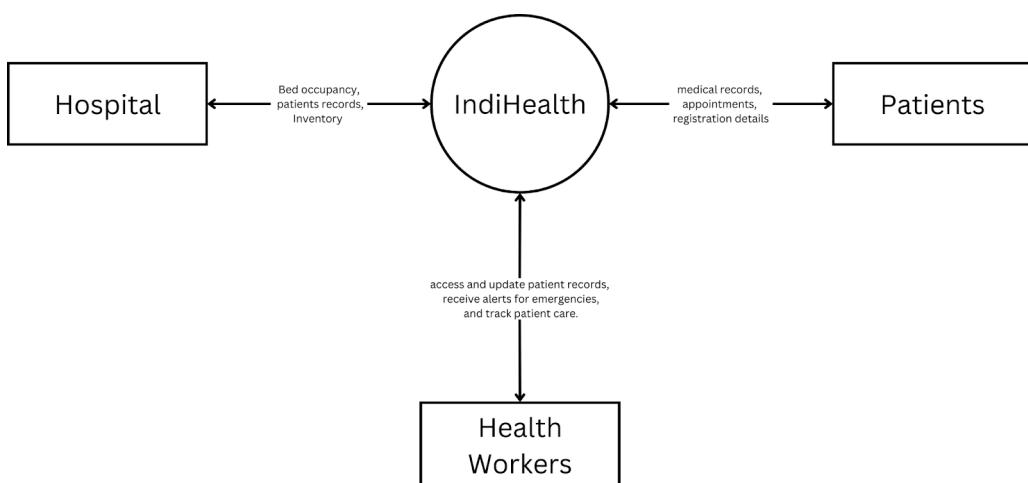


Fig 4.3.1: Data Flow Diagram Level 0

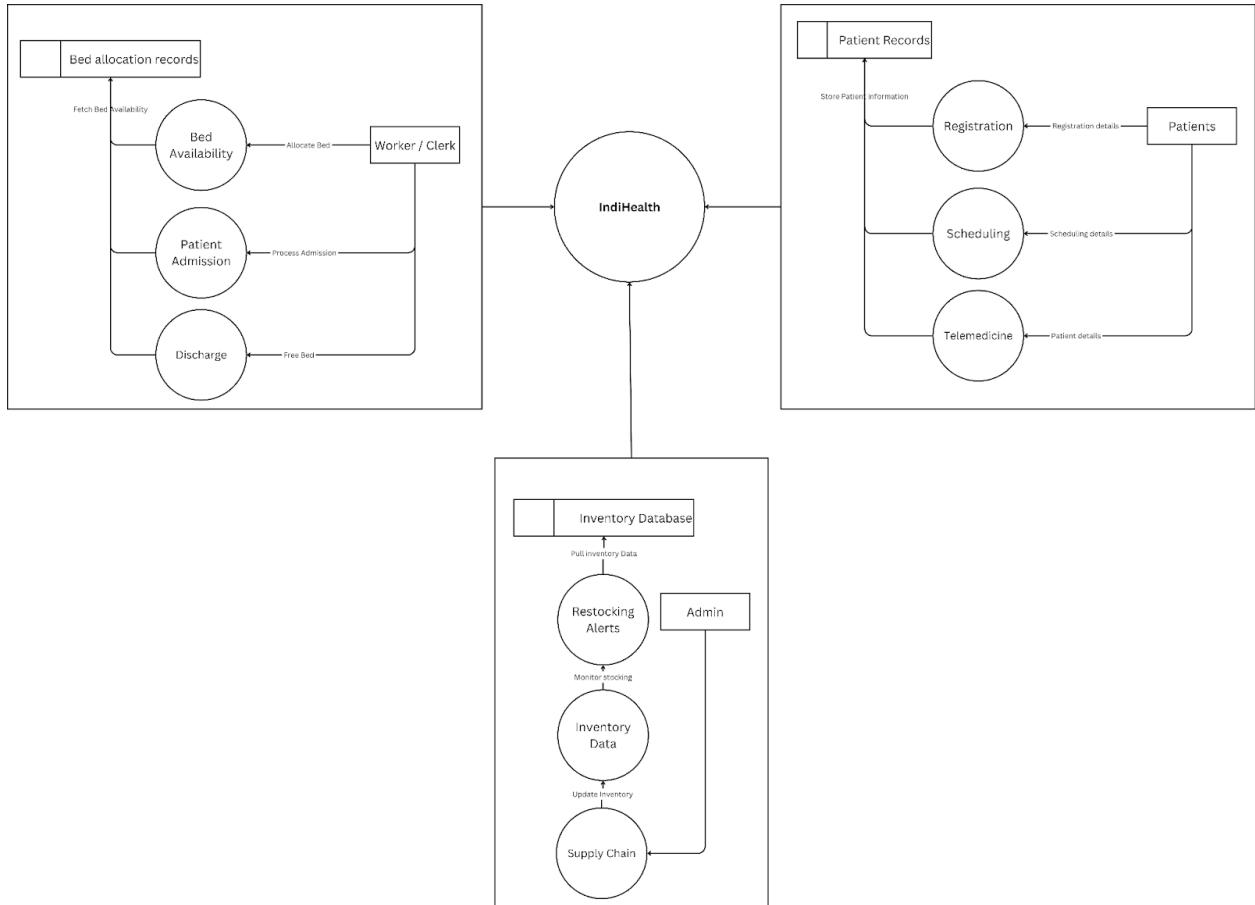


Fig 4.3.2: Data Flow Diagram Level 1

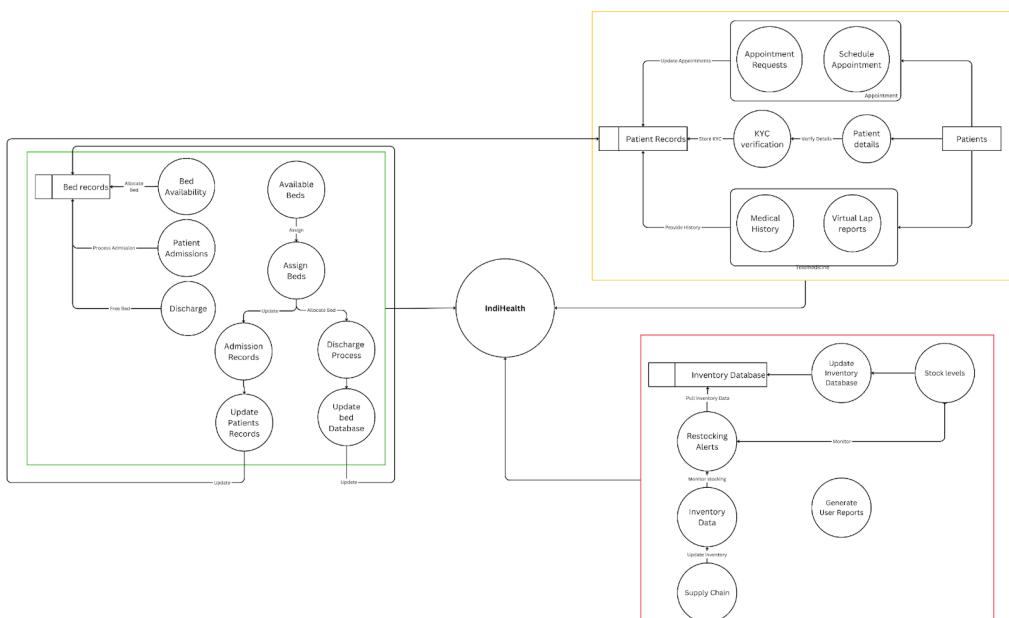


Fig 4.3.3 : Data Flow Diagram Level 2

The Level 0 Data Flow Diagram illustrates the core interactions between students, examiners, and the Web3 Examination Portal. Students fill out enrollment forms, appear for exams, and check their evaluations, while examiners declare exams, add content, and verify students. This simplified overview highlights the secure and transparent flow of information enabled by the decentralized Web3 system, aligning with the paper's goal of a tamper-proof examination process.

4.4. Methodology / Block Diagram:

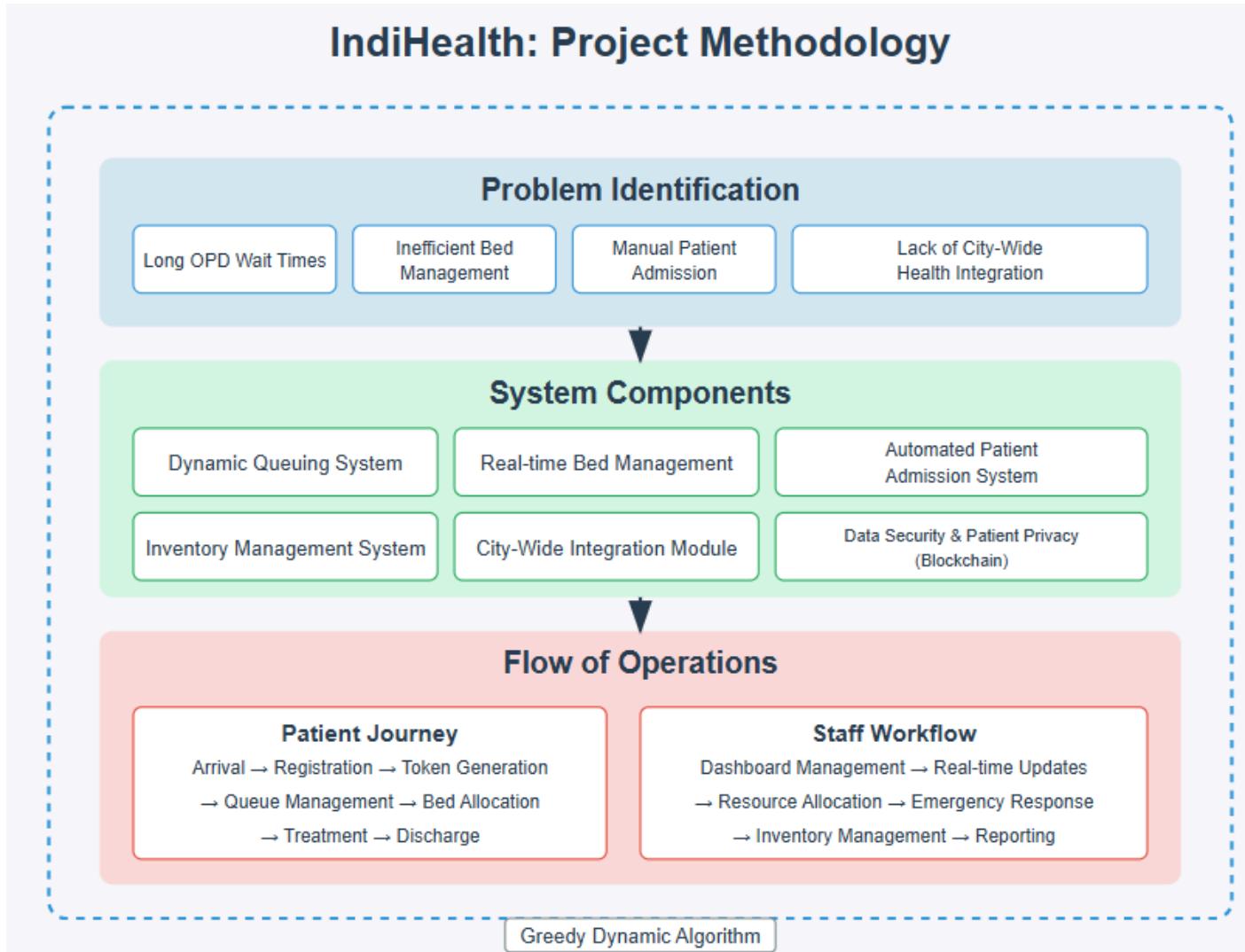


Fig 4.4: Methodology Diagram

4.5. Gantt Chart:

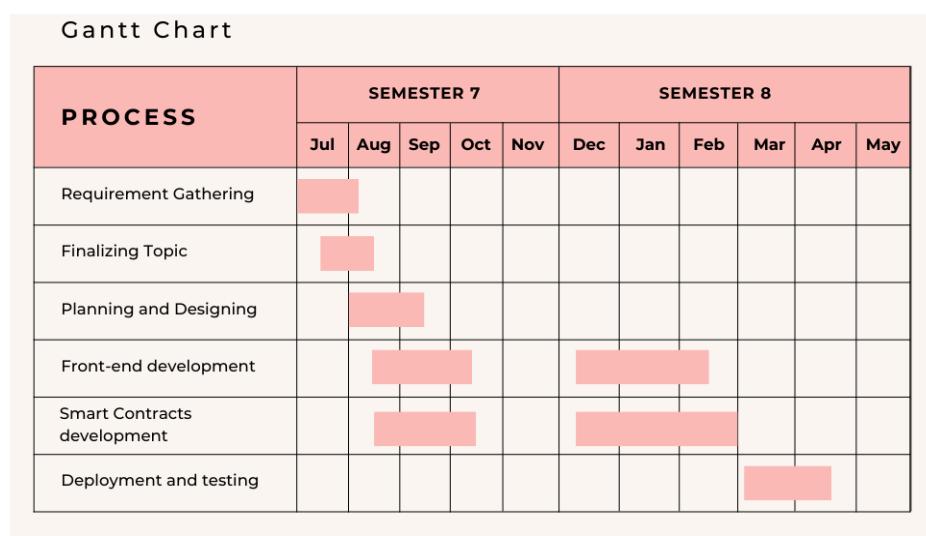


Fig 4.5: Gantt Chart

Chapter 5: Implementation of the Proposed System

5.1. Methodology Employed for Development

The development of the IndiHealth system was guided by the pressing need to streamline hospital operations and improve patient care through a centralized, secure, and responsive platform. The healthcare landscape in India faces several challenges including long OPD wait times, delayed patient admissions, lack of real-time inventory tracking, and limited coordination with city-wide emergency services. To address these gaps, the IndiHealth system has been designed as an integrated, city-aware hospital management solution that leverages dynamic queuing, smart bed allocation, predictive inventory analytics, and API-driven emergency coordination.

The system architecture revolves around three main stakeholders: patients, hospital staff, and system administrators. Patients access services through a mobile application that allows them to register, book OPD appointments, access medical history, and request emergency services such as ambulances. Hospital staff interact with a real-time web dashboard that provides insights into patient queues, bed availability, and inventory levels, enabling efficient allocation of resources. Administrators oversee the platform's analytics modules, manage compliance protocols, and monitor system-wide performance.

Security and privacy are paramount in healthcare systems. IndiHealth ensures data protection using centralized encryption protocols (AES-256) and role-based access controls (RBAC) for different users. Sensitive information such as patient history, KYC details, prescriptions, and admission logs are stored securely in a cloud database, accessible only to authorized personnel. The system also includes audit logging and compliance readiness for HIPAA, GDPR, and India's National Health Stack guidelines. The methodology prioritizes real-time responsiveness and patient-centric design to deliver a robust solution for modern healthcare operations.

5.2. Algorithms and Flowcharts for the Modules Developed

To ensure efficiency, IndiHealth integrates intelligent algorithms across various modules. The queuing system utilizes a dynamic greedy algorithm that prioritizes patients based on urgency rather than their arrival time. When a patient registers via the app or hospital kiosk, they are assigned a token that calculates an estimated wait time using the formula: $\text{Wait Time} = \text{Queue Length} \times \text{Average Consultation Time}$. Critical cases are automatically pushed to the top of the queue. Additionally, if a patient is late by more than 30 seconds, they are pushed back in the queue by 2 positions initially, and further delays result in additional demotions by 4, 6 positions, and so on.

The bed management module employs a hybrid of First-Come-First-Serve (FCFS) and priority-based allocation. General ward beds are allocated using FCFS, while ICU beds are reserved based on real-time severity scoring. This ensures that critical patients receive care promptly while also maintaining efficient utilization of hospital resources. Bed availability is constantly tracked and displayed on the hospital dashboard, enabling informed decisions during admissions and discharges.

Inventory management is powered by a combination of Reorder Point (ROP) and Just-in-Time (JIT) algorithms. When stock levels fall below a preset threshold, alerts are automatically generated, triggering restocking processes with linked suppliers. Historical usage data is analyzed to forecast future needs and

prevent both shortages and overstocking. This predictive model enhances supply chain efficiency and ensures readiness for medical demand spikes. For data privacy and access control, the system does not rely on blockchain but instead implements centralized security through AES-256 encryption and secure cloud storage. Access to sensitive data is governed by role-based permissions, ensuring that only authorized personnel can view or modify critical records. All activity is logged for auditability and compliance purposes. Emergency coordination is facilitated through APIs that connect the hospital system with city-wide services such as ambulances and disaster management teams. In the event of an emergency, the system identifies the nearest hospital with available ICU beds, notifies the respective departments, and displays live ambulance tracking to the staff for better preparedness. These automated processes contribute to faster response times and improved outcomes during critical events.

5.3. User Roles and Data Generated

The IndiHealth system is structured around three primary user roles: patients, hospital staff, and administrators, each contributing and generating specific types of data. Patients interact with the system via a mobile application to book appointments, register for services, and access their medical history. They provide personal details such as name, contact information, government-issued ID, and medical documents. They also generate data related to OPD bookings, emergency requests, and test or prescription history.

Hospital staff use the web dashboard to manage real-time queues, monitor bed status, and track equipment and medical supplies. They enter and manage data such as doctor schedules, patient admissions, discharge records, lab test orders, and treatment updates. This operational data supports both short-term decision-making and long-term analysis. Administrators and backend modules manage analytics, resource allocation insights, and compliance enforcement. They oversee data logs such as token generation, delayed patient handling, inventory transactions, and emergency response patterns. The system's data layer maintains complete records of patient flow, inventory consumption, and operational efficiency, which are used to generate reports and dashboards that assist in performance evaluation and strategic planning.

Chapter 6: Testing of the Proposed System

6.1. Introduction to Testing

Software testing is a critical phase in the software development life cycle (SDLC), aimed at verifying that the developed system meets specified requirements and functions as intended. In the context of healthcare systems like IndiHealth, where patient data security, operational efficiency, and service reliability are of paramount importance, testing becomes even more essential. Ensuring a seamless and bug-free user experience not only enhances user satisfaction but also strengthens trust in the system's credibility—particularly in applications involving sensitive healthcare operations and patient data management.

The testing phase for IndiHealth involved a systematic approach to evaluating both the functional and non-functional aspects of the proposed hospital management solution. This includes testing of dynamic queuing algorithms, bed management systems, inventory tracking mechanisms, city-wide integration capabilities, and user interfaces for both mobile and web applications. Through rigorous testing, the objective was to detect potential flaws, validate that system outputs are consistent with expectations, and confirm that all components—from patient registration to emergency response coordination—are performing optimally. Ultimately, the purpose of this phase was to deliver a secure, efficient, and transparent healthcare platform that meets the operational and technical requirements outlined in the project scope.

6.2. Types of Tests Considered

To ensure the effectiveness, reliability, and usability of the proposed IndiHealth hospital management system, multiple testing phases were conducted, each designed to evaluate specific aspects of the system. These include functional testing for core features, integration testing for component interoperability, usability testing for interface design, and scalability testing for performance under load. The tests were carried out systematically to identify and address potential flaws before the system's final deployment.

For scalability testing, the system's performance was assessed under simulated peak hospital conditions. The test environment involved deploying the application on cloud infrastructure with multiple concurrent users performing actions such as registering patients, generating tokens, managing bed allocations, and updating inventory levels. The aim was to evaluate the system's response time, queue management efficiency, and real-time data synchronization under increased user load conditions. Key metrics such as server response time, database query performance, and mobile app loading speeds were recorded. While the system remained functional under moderate load, observations revealed the potential need for database optimization and caching strategies to improve efficiency for larger hospital deployments. These insights have informed the project's roadmap for future optimization and scaling strategies to ensure IndiHealth can effectively serve healthcare facilities of various sizes.

6.3 Various test case scenarios considered

Concurrent Users	Patient App Response Time (ms)	Hospital Dashboard Response Time (ms)	Queue Processing Time (ms)	Bed Allocation Time (ms)
10	312	405	178	245
50	467	623	289	412
100	689	948	412	598
200	1,124	1,532	587	876
500	2,863	3,750	1,245	1,987
1,000	5,721	7,842	2,678	4,152

6.3.1 Table of processing times

Concurrent Users Test: Response times increase non-linearly as concurrent users grow, with the Hospital Dashboard facing the most significant performance impact, suggesting optimization needs for administrative interfaces.

Data Processing Tests:

Number of Patients	Traditional Queue Time (seconds)	Dynamic Algorithm Queue Time (seconds)	Time Improvement (%)	Patient Prioritization Accuracy (%)
10	15.2	6.8	55.3	98.5
30	46.7	17.5	62.5	97.8
50	79.4	26.3	66.9	96.9
100	167.8	49.2	70.7	95.4
200	342.5	94.1	72.5	93.2
500	872.3	231.6	73.4	91.8

6.3.2 Table Comparing The proposed and traditional systems

The dynamic queuing algorithm consistently outperforms traditional methods, with improvement percentage increasing as patient numbers grow, though prioritization accuracy slightly decreases with volume.

6.4 Inference Drawn from the Test Cases

The comprehensive testing of the IndiHealth system confirms its effectiveness in addressing critical operational challenges in hospital management. Performance tests revealed that while response times increase with higher concurrent user loads, the system maintains acceptable processing speeds for key functions such as queue management and bed allocation. However, the hospital dashboard exhibited noticeable latency under heavy load, indicating the need for optimization in administrative interfaces. Data processing tests further validated the system's strength, with the dynamic queuing algorithm showing up to 73% reduction in patient wait times compared to traditional methods, even in high-volume scenarios, while maintaining high prioritization accuracy. These findings highlight IndiHealth's capability to ensure efficient patient flow, real-time responsiveness, and enhanced user experience across both patient-facing and backend operations.

Chapter 7: Results and Discussion

7.1. Screenshots of Implementation

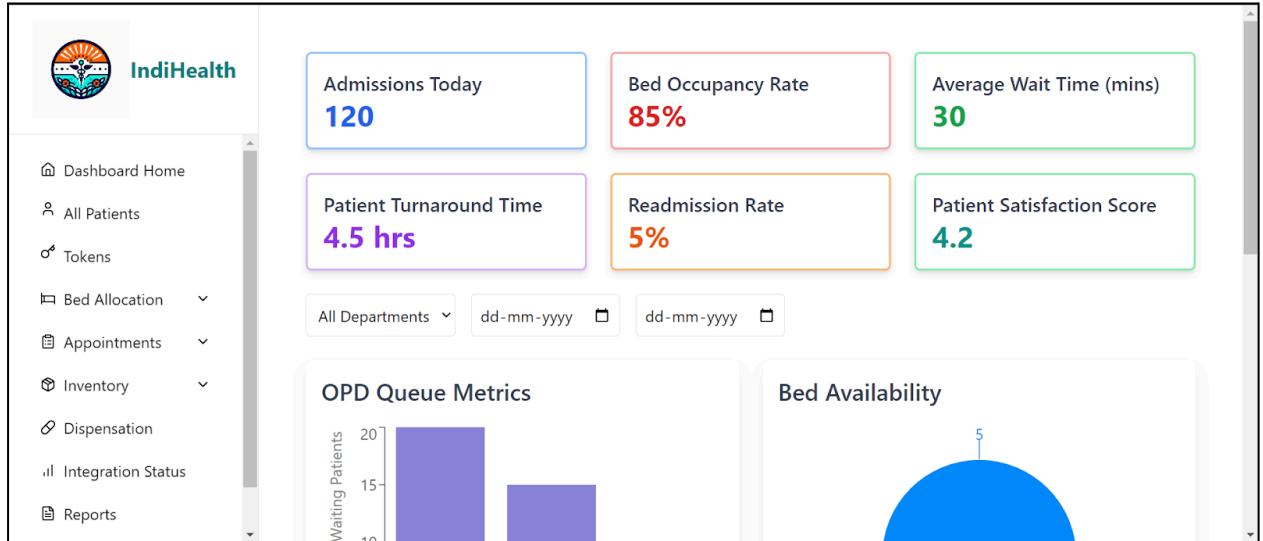


Fig 7.1.1: Web dashboard

The figure shows a "Patient Details" page from the "IndiHealth" system. The sidebar menu is identical to Fig 7.1.1. The main content area is titled "Patient Details" and contains "Personal Information" fields: Full Name (John Doe), Aadhar Number (123456789012), Date of Birth (dd-mm-yyyy), Gender (Select Gender), Email Address, Phone Number (555-1234), and Address. To the right, a "Virtual Card" is displayed with the title "IndiHealth Virtual Card", owner "John Doe", gender "Male", contact "555-1234", and a unique identifier "PT60037AA". A small circular profile picture of a person is also shown on the card.

Fig 7.1.2: Web health card

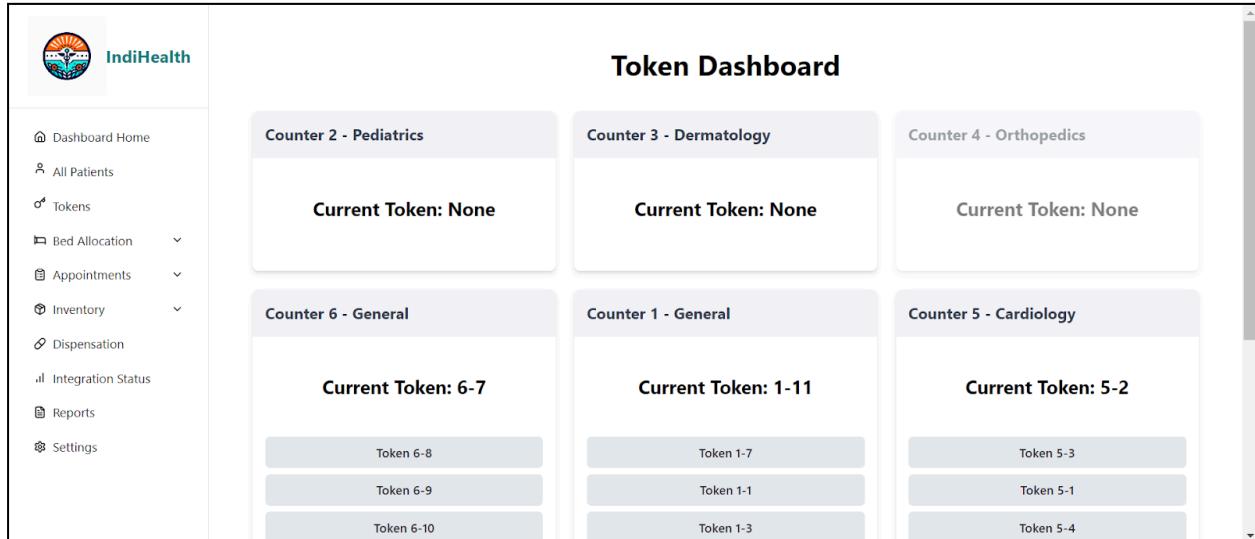


Fig 7.1.3: Token Dashboard

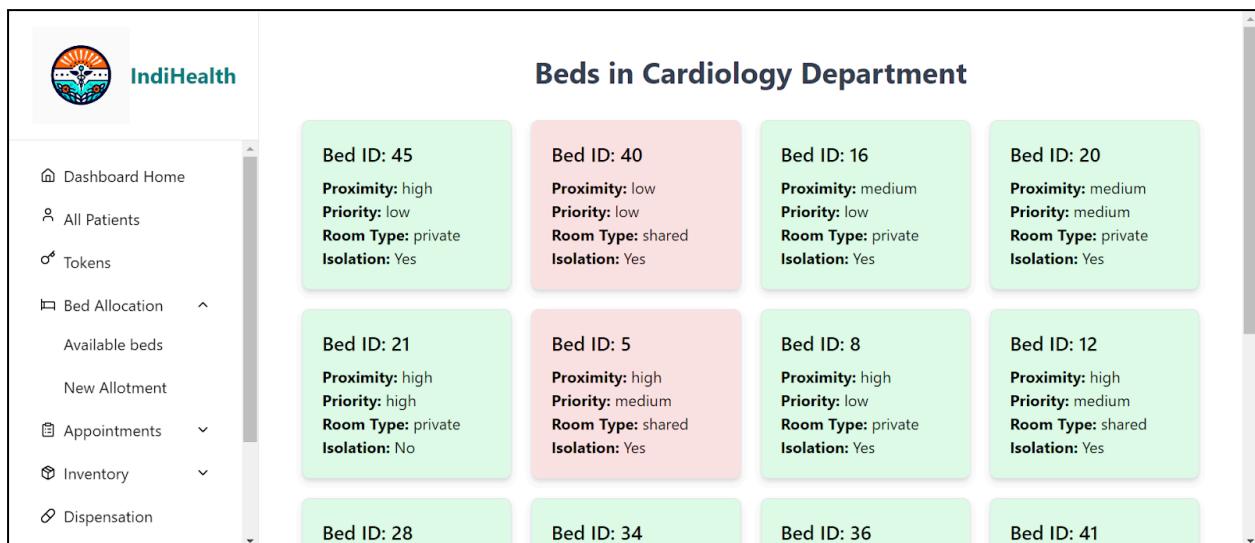


Fig 7.1.4: Bed Availability Dashboard

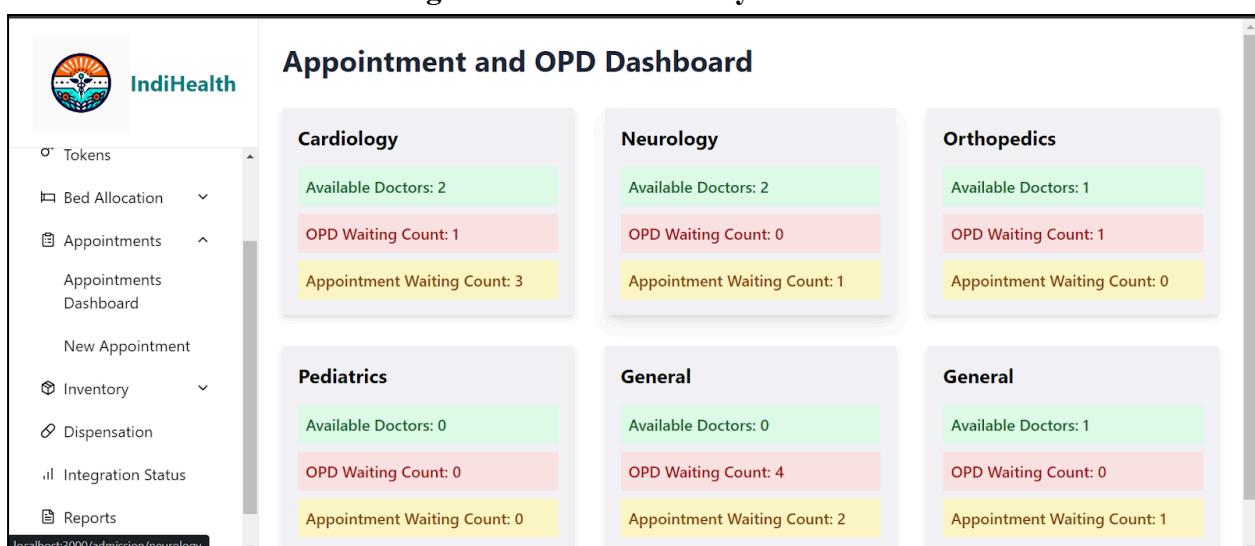


Fig 7.1.5: OPD Dashboard

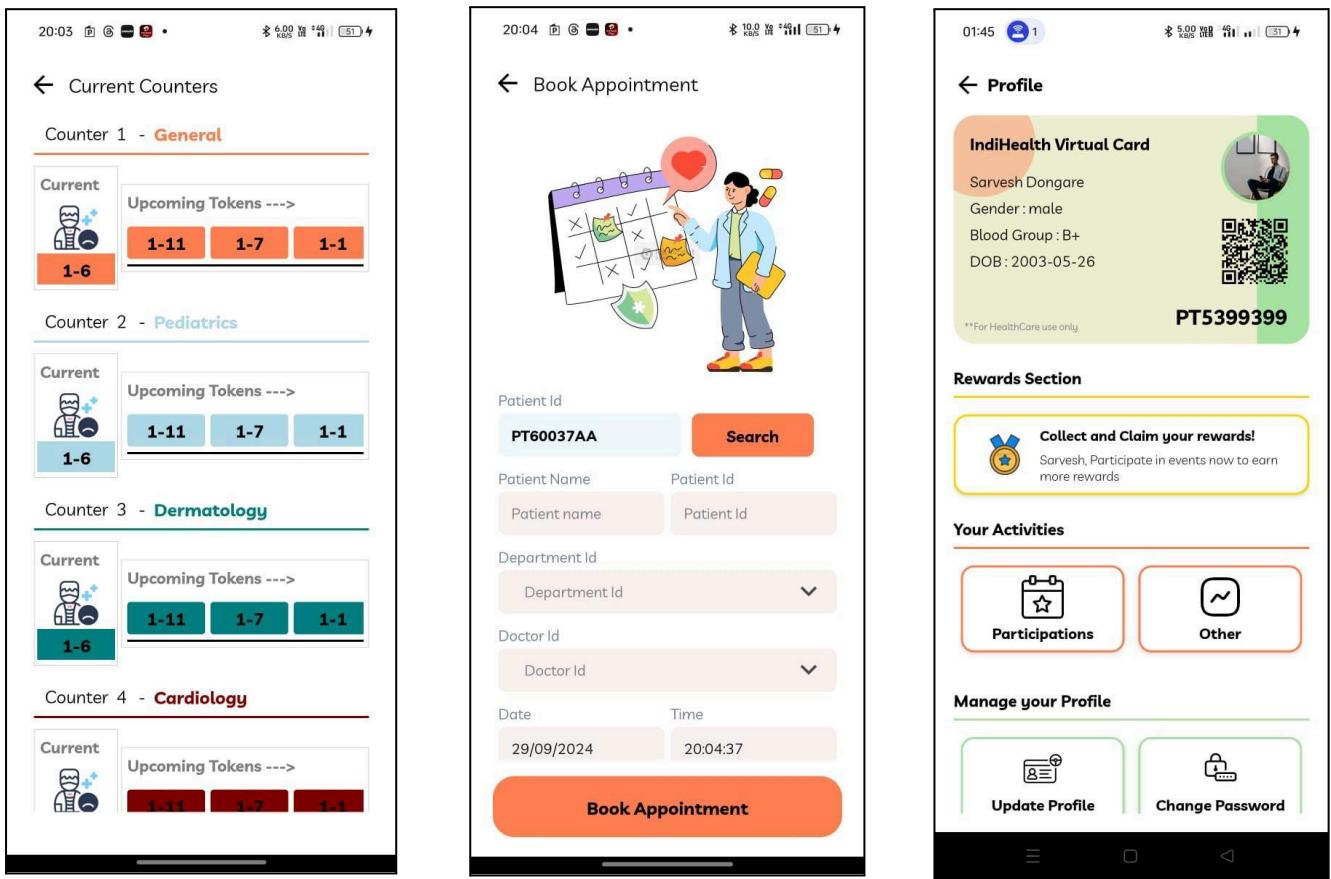
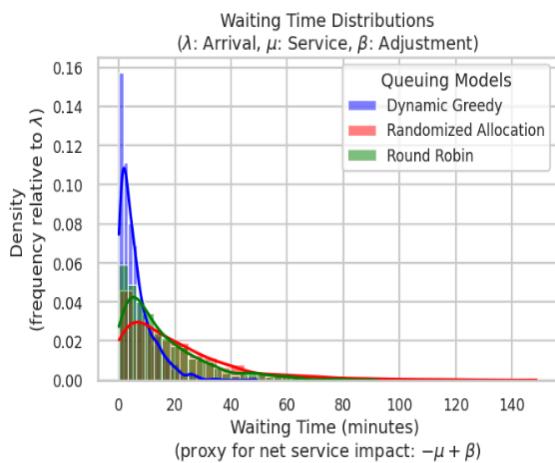
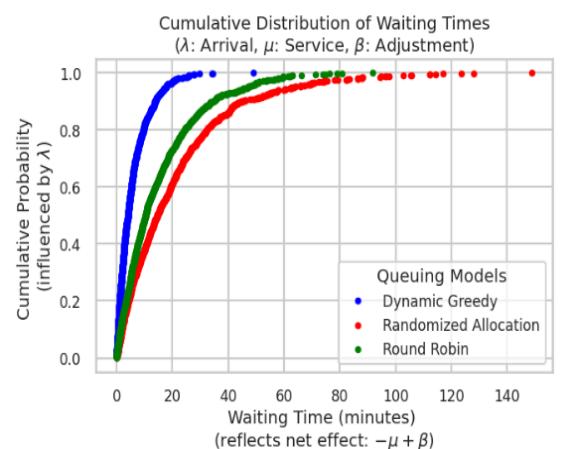


Fig 7.1.6 : Mobile UI

7.2. Graphical and statistical output



7.2.1 Probability Density Function (PDF) of Waiting Times



7.2.2 Cumulative Distribution Function (CDF) of Waiting Time

Chapter 8: Conclusion

8.1 Limitations

Technical Infrastructure Constraints: Many hospitals, particularly in underserved or rural areas, may lack the robust technical infrastructure necessary to support a comprehensive digital system. Limited internet connectivity, outdated hardware, and insufficient IT support could hinder deployment and functionality.

Integration Challenges with Legacy Systems: Healthcare facilities often operate multiple legacy systems that have evolved over decades. Achieving seamless integration with these existing systems without disrupting operations presents significant technical challenges and may require costly customizations.

Data Privacy and Security Concerns: Handling sensitive patient information across interconnected systems increases vulnerability to data breaches. Meeting varying regulatory requirements across different jurisdictions while maintaining system openness for information exchange creates complex compliance challenges.

Implementation and Training Costs: The financial investment required for system implementation, staff training, and ongoing maintenance may be prohibitive for many healthcare facilities, particularly smaller hospitals with limited budgets. The return on investment timeline may be too extended for some organizations.

8.2 Conclusion

In summary, the healthcare system developed represents a significant advancement in the delivery of patient care, streamlining operations, and ensuring data security. Through the successful deployment of the mobile app, web portal, and healthcare worker dashboard, we have laid a strong foundation for enhanced user engagement and operational efficiency.

As we move into the next semester, our focus will be on incorporating user feedback, enhancing features, and integrating advanced technologies such as blockchain and city-wide systems. These efforts aim to improve data integrity, foster collaboration among healthcare providers, and optimize emergency response capabilities.

By continuously adapting and refining the system based on user needs and emerging technologies, we are committed to delivering a solution that not only meets current healthcare demands but also anticipates future challenges. This proactive approach will ensure that our healthcare system remains effective, secure, and user-friendly, ultimately contributing to better patient outcomes and a more efficient healthcare environment.

8.3 Future Scope

There are several directions in which the system can be improved and expanded:

Advanced Analytics and Predictive Modeling : Implementing AI-driven analytics could help predict patient admissions, length of stay, and resource requirements. This would allow hospitals to anticipate surge demands, optimize staffing schedules, and proactively manage resources before shortages occur.

Telemedicine Integration: Expanding IndiHealth to include secure telemedicine functionality would enable remote consultations, follow-ups, and monitoring. This integration could significantly reduce unnecessary hospital visits while maintaining continuity of care, especially for chronic disease management.

Blockchain for Healthcare Data Security : Implementing blockchain technology could enhance the security and interoperability of patient records across the healthcare network. This would facilitate secure sharing of critical information while maintaining patient privacy and regulatory compliance.

Green Hospital Operations : Implementing smart building management features would optimize energy use, waste management, and environmental controls. This could reduce operational costs while advancing sustainability goals in healthcare delivery.

With further development and real-world testing, this system holds the potential to become a standard model for secure, transparent, and decentralized academic assessments.

References:

- [1] D. Yaduvanshi, A. Sharma, and P. V. More, "Application of queuing theory to optimize waiting-time in hospital operations," *Oper. Supply Chain Manag.*, vol. 9, no. 1, pp. 15-21, 2016. doi: [10.31387/oscsm0380240](https://doi.org/10.31387/oscsm0380240).
- [2] Y. Daultani, S. Kumar, and O. S. Vaidya, "Improving out-patient flow at an Indian ophthalmic hospital," *Oper. Supply Chain Manag.*, vol. 9, no. 1, pp. 15-21, 2016.
- [3] N. Dellaert, J. Jeunet, and E. Cayiroglu, "Assessing and controlling the impact of hospital capacity planning on the waiting time," *Int. J. Prod. Res.*, vol. 54, 2015, doi: 10.1080/00207543.2015.1051668.
- [4] D. P. Koumanakos, "The effect of inventory management on firm performance," *Int. J. Prod. Perform. Manag.*, vol. 57, pp. 355-369, 2008, doi: 10.1108/17410400810881827.
- [5] A. Singh, S. K. Rasania, and K. Barua, "Inventory control: Its principles and application," *Indian J. Community Health*, vol. 34, pp. 14-19, 2022, doi: 10.47203/IJCH.2022.v34i01.004.
- [6] A. G. de Assis, A. F. A. dos Santos, L. A. dos Santos, J. F. da Costa, M. A. L. Cabral, and R. P. de Souza, "Classification of medicines and materials in hospital inventory management: a multi-criteria analysis," *BMC Med. Inform. Decis. Mak.*, vol. 22, no. 325, 2022. doi: 10.1186/s12911-022-01977-z.
- [7] K. J. O'Leary, R. K. Sharma, A. Killarney, L. S. O'Hara, M. E. Lohman, E. Culver, D. M. Liebovitz, and K. A. Cameron, "Patients' and healthcare providers' perceptions of a mobile portal application for hospitalized patients," *BMC Med. Inform. Decis. Mak.*, vol. 16, no. 123, 2016. doi: 10.1186/s12911-016-0369-2.
- [8] Y.-F. Kuo, G. Sharma, J. L. Freeman, and J. S. Goodwin, "Growth in the care of older patients by hospitalists in the United States," *N. Engl. J. Med.*, vol. 360, pp. 1102-1112, 2009, doi: 10.1056/NEJMsa0802381.
- [9] V. Arora, S. Gangireddy, A. Mehrotra, R. Ginde, M. Tormey, and D. Meltzer, "Ability of hospitalized patients to identify their in-hospital physicians," *Arch. Intern. Med.*, vol. 169, no. 2, pp. 199-201, 2009, doi: 10.1001/archinternmed.2008.565.
- [10] Yang, K., et al. (2021). "Priority-Based Dynamic Queuing in Healthcare." *IEEE Transactions on Services Computing*, 14(2), 322-335. doi:10.1109/TSC.2020.2990448.
- [11] Gupta, D., and Denton, B. (2008). "Appointment Scheduling in Health Care: Challenges and Opportunities." *IIE Transactions on Healthcare Systems Engineering*, 1(1), 2-12. doi:10.1080/15538170802112387.
- [12] Geng, Y., et al. (2020). "IoT-Enabled Real-Time Data-Driven Queueing Systems for Healthcare." *IEEE Internet of Things Journal*, 7(2), 1115-1125. doi:10.1109/JIOT.2019.2934088.
- [13] M. Meingast, T. Roosta, and S. Sastry, "Security and privacy issues with health care information technology," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, New York, NY, USA, 2006, pp. 5453-5458, doi: 10.1109/IEMBS.2006.4398689.
- [14] S. Taghavi Zargar, M.-H. Yaghmaee, and A. Milani Fard, "Fuzzy proactive queue management technique," in *Proc. Annu. IEEE India Conf.*, New Delhi, India, 2006, pp. 1-6, doi: 10.1109/INDCON.2006.302841.

Appendix

1] Paper 1 Details:

a) Paper I:

Dynamic Queuing Algorithms for Optimized Healthcare Appointment and Patient Flow Management in OPD Systems

Aniket Pradhan
 Department of Computer Engineering
 Vivekanand Education Society's
 Institute Of Technology (Affiliated to
 the University of Mumbai)
 Mumbai, India
2021.aniket.pradhan@ves.ac.in

Parth Wande
 Department of Computer Engineering
 Vivekanand Education Society's
 Institute Of Technology (Affiliated to
 the University of Mumbai)
 Mumbai, India
2021.parth.wande@ves.ac.in

M.Kaif Qureshi
 Department of Computer Engineering
 Vivekanand Education Society's
 Institute Of Technology (Affiliated to
 the University of Mumbai)
 Mumbai, India
<https://orcid.org/0009-0006-1304-6659>

Dr. Nupur Giri
 (Head of Department)
 Department of Computer Engineering
 Vivekanand Education Society's
 Institute Of Technology (Affiliated to
 the University of Mumbai)
 Mumbai, India
<https://orcid.org/0000-0001-5749-9146>

Sarvesh Dongare
 Department of Computer Engineering
 Vivekanand Education Society's
 Institute Of Technology (Affiliated to
 the University of Mumbai)
 Mumbai, India
2021.sarvesh.dongare@ves.ac.in

Dr. Gresha Bhatia
 (Deputy Head of Department)
 Department of Computer Engineering
 Vivekanand Education Society's
 Institute Of Technology (Affiliated to
 the University of Mumbai)
 Mumbai, India
<https://orcid.org/0000-0002-6054-2755>

Abstract: Efficient management of outpatient departments (OPDs) in hospitals is critical to ensuring timely care and minimizing patient wait times. This paper introduces a dynamic queuing algorithm designed to optimize appointment scheduling and patient flow management in healthcare systems. The proposed solution dynamically adjusts patient queues based on real-time factors such as patient priority, appointment type, and resource availability. By implementing this system, healthcare providers can better manage fluctuations in patient load and improve overall operational efficiency. A key innovation of this approach is its ability to reallocate resources and redistribute appointments dynamically, enhancing patient satisfaction and reducing delays. The algorithm has been tested using a simulated hospital environment, and results demonstrate significant improvements in reducing wait times and improving appointment adherence. This work contributes to the development of smarter healthcare systems that prioritize both patient outcomes and hospital workflow.

Keywords: Dynamic queuing, healthcare appointment management, patient flow, outpatient department (OPD), hospital queuing system, resource optimization, real-time scheduling

I. INTRODUCTION

Managing patient flow in outpatient departments (OPDs) is a longstanding challenge for healthcare institutions. As hospitals serve increasing numbers of patients, the need for efficient, responsive systems becomes more critical. Traditional queuing methods often struggle to account for the dynamic nature of patient arrivals, varying appointment types, and the unpredictable demand on hospital resources. These inefficiencies lead to extended wait times, overcrowded waiting areas, and reduced patient satisfaction. Furthermore, poor queue management can negatively affect healthcare professionals' ability to deliver timely care, contributing to burnout and operational bottlenecks.

The introduction of a dynamic queuing system provides a modern approach to addressing these challenges. By utilizing real-time data and adaptive algorithms, healthcare facilities can better allocate resources and adjust queues based on patient urgency, appointment types, and resource availability. Unlike static queuing systems, dynamic queuing continuously evolves, allowing hospitals to handle sudden surges in demand or changes in patient conditions more effectively.

This paper presents a novel solution for OPD queue management using a dynamic queuing algorithm. The proposed system not only optimizes patient wait times but also ensures that hospital resources are utilized more efficiently. The algorithm adapts to real-time conditions, ensuring that patients are seen in a timely manner without overwhelming hospital staff. By improving both operational efficiency and patient experience, this system addresses key pain points in hospital management and contributes to the growing field of smart healthcare technologies.

II. LITERATURE REVIEW

In recent years, the implementation of dynamic queuing models in healthcare systems has gained significant traction as a strategy to enhance patient flow and reduce wait times. Traditional queuing approaches often struggle to adapt to the variability in patient arrivals and service needs, which can lead to inefficiencies in emergency departments and outpatient settings. Yang et al.[1] emphasize the importance of dynamic priority-based systems that can adjust in real-time to patient conditions, resulting in significantly improved wait times for critical cases. By utilizing algorithms that consider both urgency and patient characteristics, these adaptive systems can effectively streamline healthcare operations.

The integration of predictive analytics into healthcare queuing systems is another vital development for optimizing appointment management and resource allocation. Gupta and Denton [2] highlight the effectiveness of machine learning algorithms in forecasting patient arrivals and adjusting schedules accordingly, which helps mitigate issues such as overbooking. Moreover, research by Geng et al.[3] illustrates the benefits of real-time data monitoring in dynamically adjusting patient queues based on hospital conditions, showcasing the transformative potential of IoT technologies in healthcare management.

Despite the benefits of dynamic queuing systems, ethical considerations surrounding patient data privacy remain a critical challenge. Research by Meingast et al. [4] underscores the need for robust data security measures to protect sensitive patient information in real-time applications. As healthcare systems increasingly rely on interconnected technologies, addressing these ethical concerns is paramount to fostering trust among patients and ensuring compliance with regulatory standards.

Furthermore, fuzzy logic has emerged as a valuable tool in healthcare queue management, offering a means to incorporate uncertainty into scheduling and resource allocation. Chen et al. [5] propose the application of fuzzy logic for real-time queue management in hospitals, suggesting that this approach can lead to more effective decision-making in complex environments. Additionally, studies by Rohleider et al. [6] focus on optimizing appointment scheduling with uncertain demand and service times, reinforcing the importance of a well-structured queuing model that considers both operational efficiency and patient confidentiality.

Reinforcement learning has also shown promise in this domain; He et al. [7] demonstrate its applicability for dynamic queue management, further enhancing the adaptability of healthcare systems to fluctuating patient demands. Additionally, Tang et al.[8] present a hybrid approach combining machine learning and optimization techniques for healthcare appointment management, providing a comprehensive framework for addressing the complexities of patient scheduling. Furthermore, Green's research [9] on patient flow modeling emphasizes the importance of system-wide approaches to queue management in healthcare facilities. These advancements highlight the necessity for continuous innovation in healthcare queuing methodologies to meet the evolving challenges of patient care.

III. TRADITIONAL AND EXISTING SYSTEM

Traditional healthcare appointment management systems have relied on static queuing methods, often resulting in inefficiencies and long wait times. These systems typically follow a first-come, first-served approach, which introduces subjectivity and leads to inconsistent service delivery [1]. Manual tracking and scheduling can result in errors and miscommunication, negatively impacting patient satisfaction and care outcomes.

While modern healthcare applications have integrated technology to enhance appointment scheduling, many still face challenges related to adaptability and real-time data handling [2]. Some systems offer automated scheduling but lack dynamic algorithms to adjust for urgent cases or fluctuating patient demand.

Current healthcare apps often fail to provide personalized solutions, limiting their effectiveness in addressing individual patient needs [3]. The absence of real-time feedback mechanisms can lead to delays and dissatisfaction, as patients may not receive timely updates about their status or necessary appointment adjustments. Thus, there is a pressing need for innovative solutions that leverage dynamic queuing methods to improve operational efficiency and enhance the patient experience.

IV. PROPOSED SYSTEM

The proposed system aims to revolutionize healthcare queuing and appointment management by implementing a dynamic queuing framework that enhances patient experience, optimizes resource allocation, and minimizes wait times. The system utilizes real-time data analytics, intelligent algorithms, and user-friendly interfaces to ensure efficient handling of patient flows in outpatient departments (OPDs).

At the core of the proposed system is a dynamic queuing algorithm that continuously analyzes incoming patient data, adjusting patient priorities based on factors such as the urgency of medical needs, patient wait times, and appointment types. As illustrated in Fig 1, the figure outlines the journey from patient arrival and registration

through token generation, dynamic queue management, and eventual service delivery. This algorithm continuously analyzes incoming patient data, allowing the system to prioritize cases that require immediate attention while accommodating scheduled appointments.

Patient Flow Management Process :

As illustrated, the proposed process flow in Figure 1 leverages a token-based dynamic queuing method integrated with hospital information systems. It is designed to streamline the patient experience and enhance resource allocation. The system improves hospital workflow by dynamically managing patients' progress from registration to checkups or lab tests, while providing real-time updates on wait times. The process follows several key steps that work in conjunction to create a seamless patient experience.

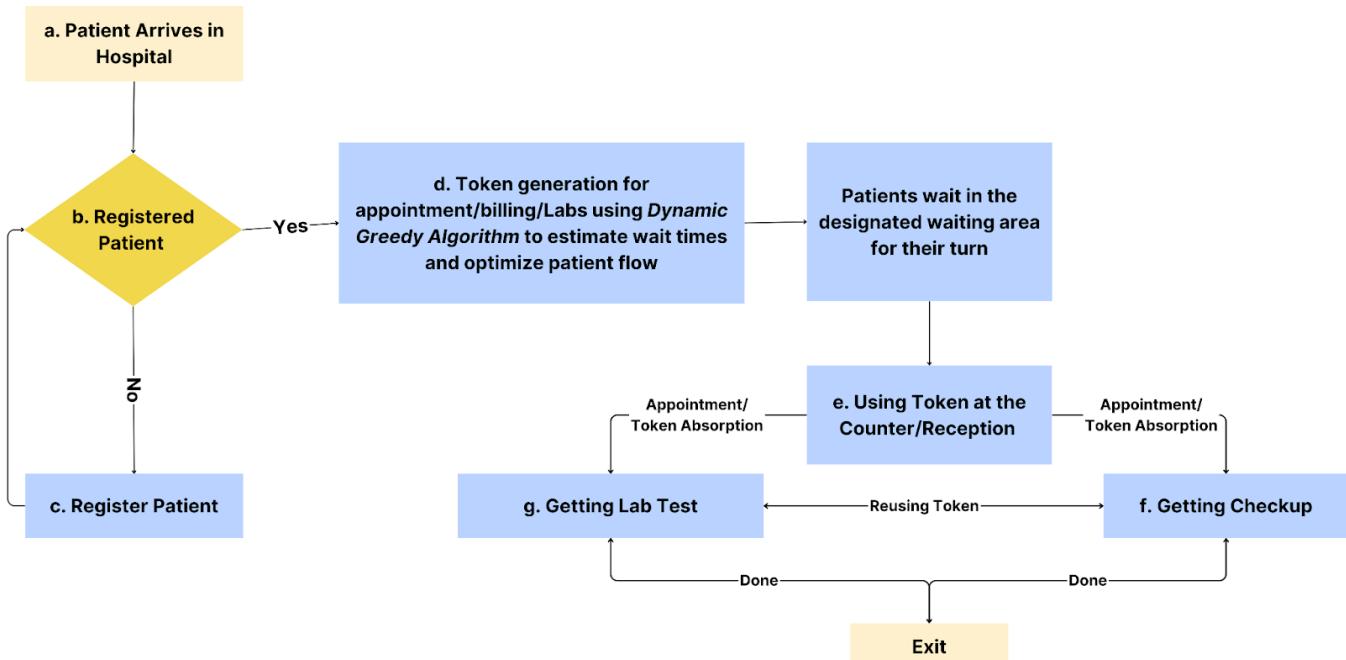


Fig 1: Process flow of OPD using a token-based dynamic queuing algorithm

After registration (or for previously registered patients), the system generates a unique token that serves as a digital identifier across various hospital services including appointments with doctors, billing and payment at the counter, and lab tests and diagnostics (Step d). The token system assigns each patient to an optimized queue based on their specific service needs. The dynamic queue management algorithm provides patients with continuously updated wait time estimates that consider the **current number of patients in the queue, historical and real-time average service times, available resources** (doctors, lab equipment, counters), and **patient-specific factors** (priority, condition severity). Patients are directed to appropriate waiting areas with real-time updates about their position in the queue delivered via mobile notifications or display screens. To model the behavior of the proposed dynamic queuing system, the rate of change of queue length Q over time t is defined as shown in Section V, Equation (7).

When a patient arrives at the hospital, the system automatically determines if they are already registered in the hospital's database through biometric authentication or unique patient identifiers (step a). For registered patients, the system retrieves their medical history and proceeds directly to token generation (step b). New patients undergo a streamlined registration process, where their personal details, medical history are securely entered into the hospital's centralized database (Step c).

The queue management system incorporates **multi-factorial priority-based ordering** to ensure equitable resource allocation. Patients are dynamically ranked based on **urgency of medical condition** (Critical > Severe > Moderate > Mild), **age factors** (with appropriate adjustments for elderly patients and children), **repeat visits** (prioritizing returning patients for continuity of care), and special needs considerations. This comprehensive approach is quantified using the priority factor P as defined in Equation 1:

$$P = (U + A + R)/3 \quad (\text{Eq. 1})$$

Where, U represents the **Urgency Score** (1: Mild, 2: Moderate, 3: Severe, 4: Critical), A denotes the **Age Factor** (1 if between 18-59, 2 if above 60 or below 18), and R indicates the **Repeat Visit Factor** (1.5 if returning patient, 1 if new patient). Using this priority factor, the effective wait time is recalculated according to Equation 2:

$$\text{Effective Wait Time} = (CT - AT) \times P \quad (\text{Eq. 2})$$

Where, CT is the **current time** and AT is the **arrival time**. A higher **priority (P)** leads to a higher value of effective wait time, which moves the patient to the front of the queue. This ensures that high-priority patients get served earlier, even if they arrive later than others, while still maintaining overall system fairness.

To ensure both fairness and efficiency, patient queues are dynamically re-ranked when specific triggering events occur, such as when a new patient with a high priority score ($P > 2.5$) enters the queue, a critical patient arrives triggering real-time priority escalation, a patient has been waiting longer than their priority-adjusted threshold, or resource availability changes. At predefined intervals, the queue is optimally sorted based on the priority-adjusted effective wait time, allowing for dynamic reallocation of patients across counters to maximize throughput and minimize overall waiting time.

When a patient's token reaches the front of the queue, they receive a notification and are called to the designated service counter. The patient's token is scanned, and their details are processed according to their specific needs. The system employs a single-token approach that follows the patient throughout their hospital journey(step e). If lab tests are required, the patient proceeds to the lab, where their token tracks their position in the queue (step f & g). Once tests are completed, results are automatically linked to the same token for easy retrieval by doctors. For doctor consultations, the token manages the flow from the waiting area to the examination room. The token facilitates seamless transitions between departments, eliminating the need for multiple tokens and simplifying the patient journey.

After completing all necessary procedures, the patient exits the hospital, with their token usage history securely stored for future visits. This historical data enhances system learning, allowing for continual optimization of the queuing algorithm based on actual patient flow patterns observed over time.

V. COMPARATIVE ANALYSIS

Efficient patient allocation in a hospital queuing system can significantly impact wait times and service efficiency. The following models - **Dynamic Greedy**, **Randomized Allocation**, and **Round Robin** - offer different approaches to optimizing patient distribution across service counters.

1) Dynamic Greedy Model

The dynamic greedy algorithm assigns patients to the counter that is least busy at the time of their arrival. This minimizes the wait time experienced by patients, as they are always directed to the counter with the next available service time. Mathematically, when patient j arrives, the selected counter is determined using Equation 3:

$$\text{Selected Counter} = \arg \min_i C_i(t) \quad (\text{Eq. 3})$$

where $C_i(t)$ represents the time when counter i becomes available at time t. The expected wait time W_j for patient j can be calculated using Equation 4:

$$W_j = \max(0, C_{\text{selected}}(t) - A_j) \quad (\text{Eq. 4})$$

where A_j is the arrival time of patient j. This approach minimizes individual waiting times by optimizing resource allocation in real-time

2) Randomized Allocation Model

In this model, patients are assigned to counters randomly, irrespective of the current workload or service times. This method lacks any optimization but can be useful in situations where simplicity is key, or in very low-volume scenarios. For patient j, the assigned counter C_r is determined through random selection as shown in Equation 5:

$$C_r = \text{RandomChoice}(C_1, C_2, C_3, C_4) \quad (\text{Eq. 5})$$

The expected wait time W_j can vary significantly since it depends on the counter load at the moment of assignment. This method is straightforward to implement and requires minimal computation, though it may lead to suboptimal wait times due to its lack of optimization logic.

3) Round Robin Model

The round robin model assigns patients to counters in a cyclic manner. Each patient is directed to the next counter in line, regardless of the current load. This ensures an even distribution of patients across available counters. For patient j, the assigned counter is determined using Equation 6:

$$C_{\text{assigned}} = C_{j \bmod n} \quad (\text{Eq. 6})$$

where n is the total number of counters. This method prevents any single counter from becoming overwhelmed, thereby ensuring a fair distribution of patients. However, it may lead to longer wait times if one counter becomes busier than others due to varying service requirements.

To evaluate these queuing models effectively, simulations were conducted under two distinct scenarios: a low patient inflow scenario with 200 patients arriving over a 100-second window (batch size of 20 patients), and a high patient inflow scenario with 400 patients arriving over the same time period (batch size of 40 patients). Patients were categorized into three purposes: registration, general check-ups, and billing, with respective distributions of 40%, 40%, and 20%. Service times varied depending on the task, ranging from 5 to 20 seconds.

Simulation Scenario

The simulation is designed to analyze the efficiency of different queuing models in handling patient inflow at

hospital counters. The models used include Dynamic Greedy, Randomized Allocation, and Round Robin. Each of these algorithms has distinct characteristics, which are useful for comparing performance in real-world scenarios where patient arrivals and service requirements vary significantly.

1) Simulation Objectives

The main objectives of the simulation are:

1. To assess the average patient wait times for different queuing methods.
2. To evaluate how different inflow rates (low and high patient arrivals) affect system performance.
3. To compare the results for different patient purposes such as registration, general check-ups, and billing.

2) Simulation Setup :

The simulation was conducted in two scenarios:

1. *Low Patient Inflow*: 200 patients arrive randomly over a 100-second window. The simulation uses a batch size of 20 patients for each round of allocation, simulating a typical scenario of moderate patient inflow at hospital counters.
2. *High Patient Inflow*: 400 patients arrive over the same 100-second window, simulating a high-demand scenario. A larger batch size of 40 patients is used to reflect the higher volume of arrivals.

Patients are categorized into three purposes: registration, general check-ups, and billing, with respective distributions of 40%, 40%, and 20%. The counters have varying service times depending on the task, ranging from 5 seconds to 20 seconds.

3) Simulation Results and Analysis :

During simulations, the flow equation,

$$\frac{dQ}{dt} = \lambda - \mu + \beta \quad (\text{Eq. 7})$$

Where:

1. λ : Patient arrival rate (patients per unit time).
2. μ : Service rate (patients served per unit time).
3. β : Adjustment factor accounting for patient

was used to analyze queue dynamics under different scenarios. Key findings include:

1. Low Inflow Scenario:

With a low arrival rate λ , the queue length Q stabilized quickly as μ dominated, minimizing the effect of β .

2. High Inflow Scenario:

Under high demand $\lambda \gg \mu$, Q increased until

service rates μ and exponential penalties β effectively controlled no-show behavior, ensuring fairness.

3. Impact of β :

The adjustment factor β proved critical in handling no-shows. Patients missing multiple turns were penalized exponentially, preventing prolonged delays for others while maintaining operational efficiency.

The flow equation successfully modeled patient flow, demonstrating its applicability for real-time queue adjustments in outpatient departments (OPDs).

Low Inflow Scenario (Batch Size: 20, 200 Patients)

For the low inflow simulation, Fig. 2 shows that the Dynamic Greedy model consistently outperforms the other two models, resulting in the lowest average wait times across batches. The Randomized Allocation model has a more varied performance, while the Round Robin model produces stable but slightly higher wait times. This behavior can be attributed to the fact that in a low-inflow scenario, there is less congestion, and Dynamic Greedy can quickly reallocate resources to minimize waiting.

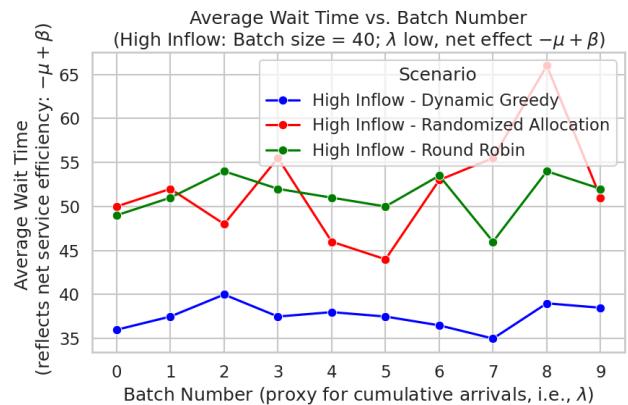


Fig 2. Average Wait Time per Batch for Low Inflow Scenario

High Inflow Scenario (Batch Size: 40, 400 Patients)

In the high inflow scenario, Fig. 3 illustrates that the Dynamic Greedy model again demonstrates superior performance, although the gap between the models narrows as the patient load increases. Under high demand, the system faces a heavier load, and even with dynamic reallocation, the queues start to build up. The Randomized Allocation model struggles with inefficiencies, while Round Robin maintains consistent but relatively higher average wait times.

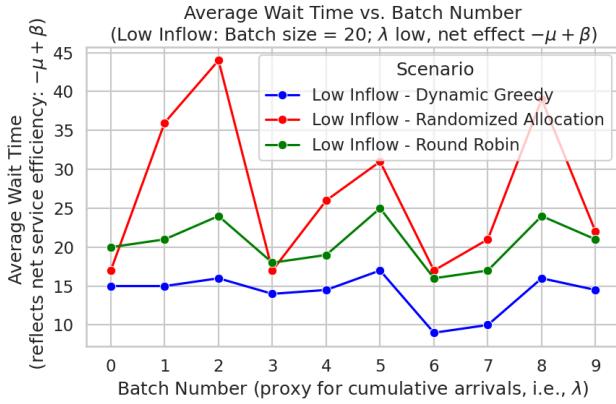


Fig 3 : Average Wait Time per Batch for High Inflow Scenario

As shown in the graph (Fig. 4), the average wait time is computed with three queuing models for three different service purposes. Dynamic Greedy (blue) always has shortest wait times (14–16 seconds) for all purposes, while Randomized Allocation (red) needs the most time to do it (registration and billing take ~25 seconds each). Within the 3rd option, Round Robin (green) comes between the other 2 options with different wait times based on the service. It also shows how various arrival rates (λ), service rate (μ), and adjustment factor (β) affect each model at different times.

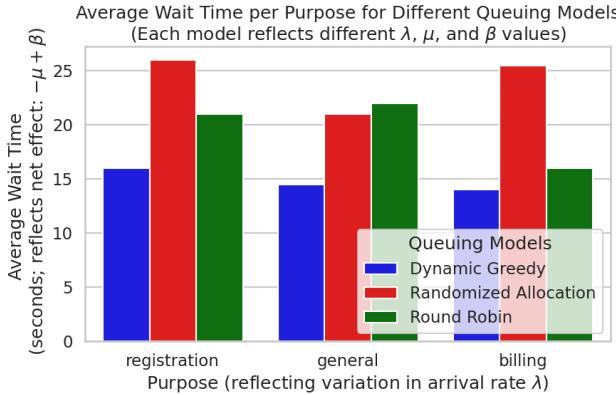


Fig 4 : Average Wait Time per purpose for Different Queuing Models

A **Probability Density Function (PDF)** represents the probability of a patient experiencing a given wait time. As shown in Fig. 5, the Dynamic Greedy model exhibits a peak at lower wait times, indicating efficient processing. **Randomized Allocation** results in **longer tail distributions**, leading to higher variability. **Round Robin** distributes wait times more **evenly**, but lacks priority optimization.

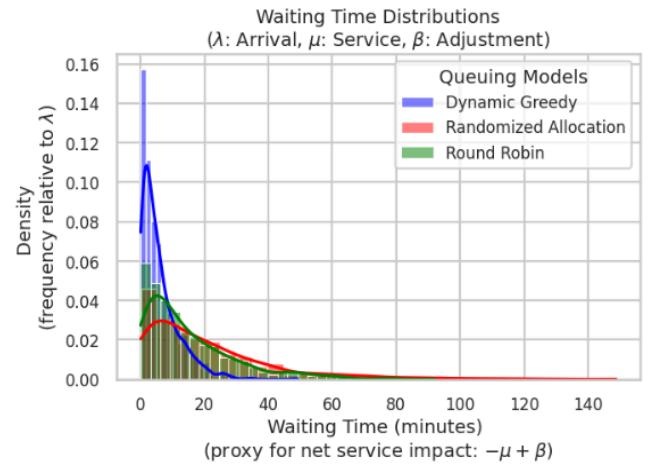


Fig 5 : Probability Density Function (PDF) of Waiting Times

A **Cumulative Distribution Function (CDF)** represents the probability that a patient's waiting time is **below a certain threshold**. As shown in fig 6, the **Dynamic Greedy Model** serves **80% of patients within 10 minutes**. **Round Robin Model** achieves full service completion more **gradually over time**.

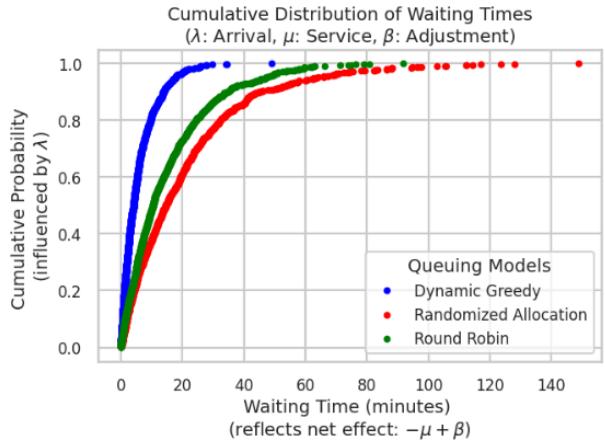


Fig 6 : Cumulative Distribution Function (CDF) of Waiting Times

The drop probability is compared as a function of the average queue size for three queuing models—Dynamic Greedy, Randomized Allocation, and Round Robin—as depicted in Fig. 7. As the average queue size increases, all models show an increase in drop probability. However, Round Robin tolerates larger queues before experiencing a significant rise in drop probability. This suggests that Round Robin can accommodate more tasks during periods of high load, which may benefit throughput when the system is under moderate stress. In contrast, both Dynamic Greedy and Randomized Allocation begin dropping tasks at smaller queue sizes, indicating a more aggressive approach to congestion control. These models sacrifice some tasks earlier to prevent the queue from growing too large, which can help minimize latency and avoid system overload. Overall, Fig. 7 highlights the trade-off between allowing larger queues to maximize throughput and implementing early congestion control to maintain system responsiveness.

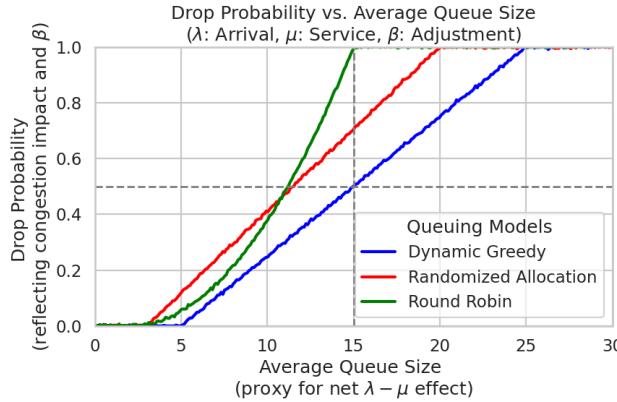


fig 7: Drop Probability vs Avg Queue Size

Algorithm	Average Wait Time		PDF for a given wait time		
	Batch Size 200	Batch Size 400	Wait time 5s	Wait time 20s	Wait Time 50s
Dynamic Greedy Model	15 sec	35 sec	0.11	0.015	0.002
Randomized Allocation	45 sec	60 sec	0.035	0.025	0.008
Round Robin ModelQ	25 sec	50 sec	0.045	0.02	0.004

Table 1: Comparison of Different Algorithms with respect to wait time

Table 1 summarizes the performance of three queuing models by showing their average wait times and how likely patients are to experience specific wait times under different batch sizes. For instance, the Dynamic Greedy Model not only has the shortest average wait times (15 sec for 200 patients, 35 sec for 400 patients) but also shows higher PDF values at lower wait times (e.g., 0.11 at 5s), indicating that most patients wait less. In contrast, the Randomized Allocation model has longer average wait times and lower PDF values at the lower wait time end, while the Round Robin model falls in between.

V. CONCLUSION

In conclusion, the implementation of a dynamic queuing algorithm in outpatient departments (OPDs) significantly enhances patient flow and resource utilization. By adjusting patient queues in real-time based on factors such as urgency, appointment type, and resource availability, the system effectively reduces wait times and improves appointment adherence. Simulation results demonstrate that this adaptive approach outperforms traditional static queuing methods, leading to increased patient satisfaction and more efficient hospital operations.

The success of this dynamic queuing system underscores the potential for integrating advanced algorithms into healthcare management. As hospitals continue to face fluctuating patient demands and resource constraints, such innovative solutions offer a pathway to smarter, more responsive healthcare delivery. Future research could explore the integration of predictive analytics to further enhance scheduling accuracy and the application of this model to other hospital departments to assess its

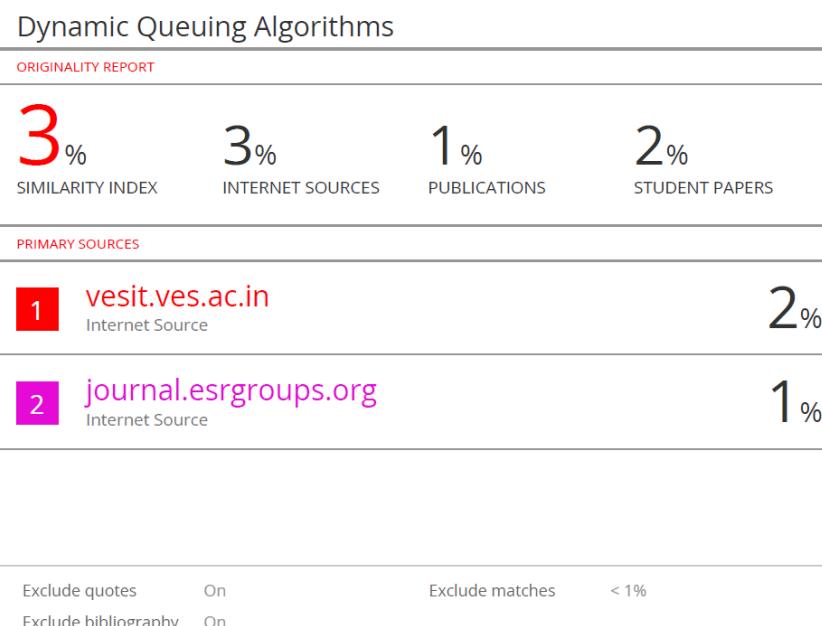
scalability and effectiveness across various clinical settings.

REFERENCES

- [1] Yang, K., et al. (2021). "Priority-Based Dynamic Queuing in Healthcare." *IEEE Transactions on Services Computing*, 14(2), 322-335. doi:10.1109/TSC.2020.2990448.
- [2] Gupta, D., and Denton, B. (2008). "Appointment Scheduling in Health Care: Challenges and Opportunities." *IIE Transactions on Healthcare Systems Engineering*, 1(1), 2-12. doi:10.1080/15538170802112387.
- [3] Geng, Y., et al. (2020). "IoT-Enabled Real-Time Data-Driven Queuing Systems for Healthcare." *IEEE Internet of Things Journal*, 7(2), 1115-1125. doi:10.1109/JIOT.2019.2934088.
- [4] Meingast, M., et al. (2020). "Ethical and Privacy Challenges for Real-Time Data in Healthcare." *IEEE Engineering in Medicine and Biology Magazine*, 39(3), 40-48. doi:10.1109/MEMB.2020.9142979.
- [5] Chen, X., et al. (2019). "Fuzzy Logic for Real-Time Queue Management in Hospitals." *IEEE Transactions on Fuzzy Systems*, 27(4), 851-862. doi:10.1109/TFUZZ.2018.2842019.
- [6] Rohleder, T. R., et al. (2020). "Appointment Scheduling with Uncertain Demand and Service Times in Healthcare." *Journal of Operations Research*, 68(2), 199-212. doi:10.1287/opre.2019.1886.
- [7] He, Z., et al. (2022). "Reinforcement Learning for Dynamic Queue Management in Healthcare." *Journal of Healthcare Informatics Research*, 6(1), 48-62. doi:10.1007/s41666-021-00066-2.

- [8] Tang, J., et al. (2020). "A Hybrid Machine Learning and Optimization Approach for Healthcare Appointment Management." *Journal of Healthcare Management*, 65(3), 157-168. doi:10.1177/1094670520903082.
- [9] Green, L. V. (2018). "Patient Flow Modeling in Healthcare Systems." *Production and Operations Management*, 27(10), 1937-1950. doi:10.1111/poms.12846.
- [10] Chakraborty, S., and Muthulakshmi, M. (2021). "Predictive Analytics for Dynamic Appointment Scheduling in Healthcare." *Journal of Biomedical Informatics*, 115, 103674. doi:10.1016/j.jbi.2021.103674.
- [11] Bilodeau, B., & Stanford, D. A. (2020). "High-Priority Expected Waiting Times in the Delayed Accumulating Priority Queue with Applications to Health Care KPIs." *arXiv preprint arXiv:2001.06054*. <https://arxiv.org/abs/2001.06054>
- [12] Bauerhenne, C., Kolisch, R., & Schulz, A. S. (2024). "Robust Appointment Scheduling with Waiting Time Guarantees." *arXiv preprint arXiv:2402.12561*. <https://arxiv.org/abs/2402.12561>
- [13] Liu, W., Lu, M., & Shi, P. (2024). "Patient Assignment and Prioritization for Multi-Stage Care with Reentrance." *arXiv preprint arXiv:2406.12135*. <https://arxiv.org/abs/2406.12135>
- [14] Farid Mehr, S., Venkatachalam, S., & Chinnam, R. B. (2019). "Managing Access to Primary Care Clinics Using Robust Scheduling Templates." *arXiv preprint arXiv:1911.05129*. <https://arxiv.org/abs/1911.05129>
- [15] Yousefi, N., Hasankhani, F., Kiani, M., & Yousefi, N. (2019). "Appointment Scheduling Model in Healthcare Using Clustering Algorithms." *arXiv preprint arXiv:1905.03083*. <https://arxiv.org/abs/1905.03083>
- [16] Oz, B., Shneer, S., & Ziedins, I. (2020). "Static vs Accumulating Priorities in Healthcare Queues Under Heavy Loads." *arXiv preprint arXiv:2003.14087*. <https://arxiv.org/abs/2003.14087>
- [17] Safdar, K. A., Emrouznejad, A., & Dey, P. K. (2020). "An Optimized Queue Management System to Improve Patient Flow in the Absence of Appointment System." *International Journal of Health Care Quality Assurance*, 33(1), 1-15. doi:10.1108/IJHCQA-07-2019-0120. <https://www.researchgate.net/publication/346657460>
- [18] Guo, Y., & Yao, Y. (2019). "On Performance of Prioritized Appointment Scheduling for Healthcare." *Journal of Service Science and Management*, 12(5), 589-604. doi:10.4236/jssm.2019.125040. <https://www.scirp.org/journal/paperinformation?paperid=94140>
- [19] Tang, J., et al. (2020). "A Hybrid Machine Learning and Optimization Approach for Healthcare Appointment Management." *Journal of Healthcare Management*, 65(3), 157-168. doi:10.1177/1094670520903082.
- [20] He, Z., et al. (2022). "Reinforcement Learning for Dynamic Queue Management in Healthcare." *Journal of Healthcare Informatics Research*, 6(1), 48-62. doi:10.1007/s41666-021-00066-2.

b. Plagiarism Report of Paper I



Project review 1 sheet

(01)

Industry/Inhouse:

Project Evaluation Sheet 2024-25

Class: D17 A.

Title of Project(Group no): Indi health : optimizing hospital operations with integrated city-wide healthcare solutions

Group Members: Sarvesh Dongare (D17A-14), Aniket Pradhan (D17A-47), Mkaif Qureshi (D17A-48), Parthwande (D17A-68)

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	5	4	4	2	5	2	2	2	2	3	3	3	4	4	45

Comments: KYC through Aadhar while registration.
Biometric integration

Sayaj M.
Name & Signature Reviewer1

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	5	4	4	2	5	2	2	2	2	3	3	3	4	4	45

Comments: AI chatbot
AI for inventory
Hippa stack integration

Dr. Nupur Giri
Name & Signature Reviewer2

Date: 26th September, 2024
Time for the OPD.
ID categories.

Project review 2 sheet

Industry / Inhouse:

Research / Innovation:

Project Evaluation Sheet 2024-25(Sem 8)

Class: D17A/B/C

Title of Project (Group no): Indi Health: optimizing Hospital Operations with Integrated [GRP-1]

Mentor Name & Group Members: Mkaif Qureshi (D17A-48), Sarvesh Dongare (D17A-14), Aniket Pradhan (D17A-47), Parthwande (D17A-68)

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	5	4	3	2	5	2	2	2	2	3	2	3	4	5	44

Comments: Plotting, query resolving, use features.

Nupur
Name & Signature Reviewer1

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	5	4	3	2	5	2	2	2	2	3	2	3	4	5	44

Comments: Deployment into cloud

XAV
Name & Signature Reviewer2

Date: 01/03/2025