

# **Government Healthcare System: A Centralized Desktop Application for Government Hospital Management Using Python, Kivy, PostgreSQL and Machine Learning**

Prof.Pudale A. H., R. A.Sawant, Todkar V. U., Patil V. S., Shinde S. B.

(Department of Computer Science and Engineering, Dr. J. J. Magdum College of Engineering, Jaysingpur, Maharashtra, India  
Under the Guidance of Prof. A.H.Pudale)

\*\*\*\*\*

**Abstract**—The Government Healthcare System (GHS) is a centralized desktop application developed using Python (Django backend), Kivy (GUI frontend), PostgreSQL (database), and scikit-learn (Machine Learning) to digitize and streamline government hospital operations. The system provides role-based access for six user types: Admin/DHO, Doctor, Patient, ASHA Worker, Inventory Manager, and Government Authority. Key features include a real-time District Authority Dashboard with comprehensive operational metrics, an 8-module colour-coded Doctor Dashboard, inter-facility patient referral management with Emergency/Urgent/Routine priority and real-time status tracking, Government Scheme integration (Ayushman Bharat), ML-based disease prediction, pharmacy inventory management, community health tracking, and automated SMS/Email notifications. Testing results demonstrate a 100% pass rate across all 28 test cases, confirming the system’s reliability and readiness for deployment in government healthcare facilities in districts such as Sangli and Kolhapur.

**Keywords** — Government Healthcare System, Hospital Management System, Python Django, Kivy, PostgreSQL, Machine Learning, RBAC, Patient Referral, Ayushman Bharat, scikit-learn.

## **I. I. INTRODUCTION**

The healthcare sector is a vital component of public welfare, especially in developing regions where government hospitals serve a large portion of the population. In districts such as Sangli and Kolhapur, government healthcare institutions play a major role in providing affordable and accessible medical services to both urban and rural populations. However, increasing patient load, limited infrastructure, and reliance on manual or semi-digitized systems often lead to inefficiencies, delays, and difficulties in managing patient records.

Traditional methods of maintaining healthcare data result in issues such as data duplication, loss of records, and long waiting times. These challenges affect the quality of healthcare services and create difficulties for both patients and healthcare professionals. With advancements in technology, there is a growing need to adopt digital solutions

that can improve efficiency, accuracy, and accessibility in healthcare systems.

This project focuses on developing a Government Healthcare Management System using Python, Django, Kivy, and PostgreSQL, with an integrated Machine Learning module (scikit-learn) for disease prediction. The system aims to provide a centralized platform for managing patient information, appointments, and hospital operations. It is designed to be user-friendly, secure, and scalable, ensuring efficient handling of healthcare services.

### **A. Problem Statement**

Government healthcare institutions face several challenges due to the lack of an integrated system. Manual record-keeping leads to data loss, duplication, and inefficiencies in managing patient information. Doctors often do not have quick access to patient history, which affects diagnosis and treatment. Additionally, appointment scheduling is not properly managed, resulting in overcrowding and long waiting times. Inter-facility patient referrals are managed manually with no status tracking, and district health officers lack real-time operational data for informed governance.

### **B. Objectives**

The key objectives of the GHS are: (1) To develop a centralized desktop application integrating all government hospital operations into a single role-based platform. (2) To provide the District Health Officer with a real-time dashboard showing comprehensive operational metrics. (3) To implement inter-facility patient referral with Emergency/Urgent/Routine priority and real-time status tracking. (4) To integrate Machine Learning (scikit-learn) for disease prediction and health trend analytics. (5) To ensure data security through Role-Based Access Control (RBAC) for all six user types.

## **II. II. LITERATURE REVIEW**

The healthcare sector has witnessed significant transformation over the past few decades due to rapid advancements in information technology. Hospital Management Systems (HMS) have evolved from simple record-keeping tools to comprehensive digital platforms [6], [7]. Various researchers have studied the implementation of HMS in different healthcare settings. Studies by Tiwari et al. [3] emphasize the importance of integrating multiple hospital modules into a unified platform to improve operational efficiency and reduce redundancy. Research by Kar et al. [2] specifically addresses the challenges faced by Indian government hospitals such as overcrowding, long waiting times, and lack of real-time data visibility, proposing technology-driven solutions to streamline patient flow and enhance transparency. Role-Based Access Control (RBAC) has been widely adopted in healthcare information systems to enforce data security and privacy [8]. PostgreSQL has been demonstrated as a reliable and scalable relational database suitable for healthcare data management [9]. Django's REST framework has been used effectively in building secure backend APIs for medical applications [10].

Kumaran et al. [1] explored advanced features of modern HMS including real-time data access, mobile application support, and integration with emerging technologies. Yulianti and Muhardi [4] demonstrated the effectiveness of the Structured Design Life Cycle (SDLC) methodology in HMS development, showing how structured processes reduce development risks and ensure system scalability. Machine Learning techniques including Random Forest and Support Vector Machines have been successfully applied for disease prediction in clinical settings [11], [12]. The Kivy framework has been shown to be effective for building cross-platform GUI applications in Python [13]. The Ayushman Bharat Pradhan Mantri Jan Arogya Yojana (PM-JAY) scheme has been studied as a critical digital health intervention for economically weaker sections of Indian society [14]. ASHA workers play a crucial role as community health facilitators in rural India, and digital tools for tracking their activities have been identified as a key need [15]. Most existing solutions do not adequately address inter-facility patient referral management with priority tracking, district-level real-time operational dashboards, integration with government schemes like Ayushman Bharat, ASHA worker community health tracking, or ML-based

disease prediction. The GHS has been designed to bridge these gaps.

### III. III. METHODOLOGY

The GHS is developed using a modular SDLC approach [16]. The system runs as a desktop application on government hospital computers. The Kivy frontend renders role-specific screens [13]. The Django backend handles all business logic, database operations (PostgreSQL) [9], ML predictions (scikit-learn) [11], authentication, and external service integrations. Nine core modules are integrated through a centralized PostgreSQL database and local Django REST API layer [10] with Role-Based Access Control (RBAC) [8] enforced at every screen boundary.

#### A. System Architecture

The GHS uses a three-tier layered architecture [17]. The Presentation Layer is built using Kivy GUI framework [13] and renders role-specific desktop application screens. The Application Layer is implemented using Django (Python) [10] and scikit-learn for Machine Learning [11], comprising Authentication and RBAC [8], Patient Service, Appointment Service, Doctor/Clinical Service, Referral Service, Inventory Service, ASHA Worker Service, Government Scheme Service, ML Prediction Service, and Notification Service. The Data Layer is a PostgreSQL relational database [9]. External Services include SMS Gateway, Django SMTP Email Server, Government Health Portal API for Ayushman Bharat [14], and Aadhaar Verification Service [18].

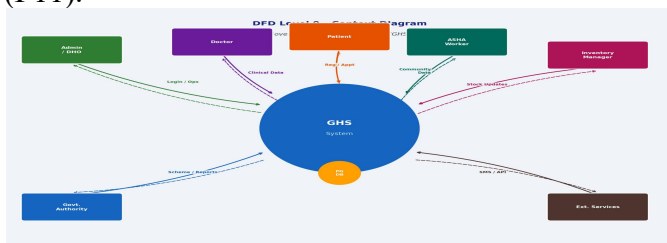
#### B. Modules

The GHS comprises nine integrated modules. (1) Admin Module: Provides the District Authority Dashboard with real-time metrics covering Staff, Patients, Appointments, Lab, Finance, Schemes, and Community. (2) Doctor Module: 8-module colour-coded clinical dashboard. (3) Patient Module: Registration, EMR, prescriptions, lab reports, scheme enrollment, and referral initiation. (4) Government Scheme Module: Ayushman Bharat integration with eligibility verification. (5) Inventory Module: Medicine stock, expiry monitoring, and automated alerts. (6) Employee Module: Staff records, attendance, and payroll management. (7) ASHA Worker Module: Community health tracking, immunization, and health camps. (8) Appointment Module: OPD booking with real-time availability and SMS

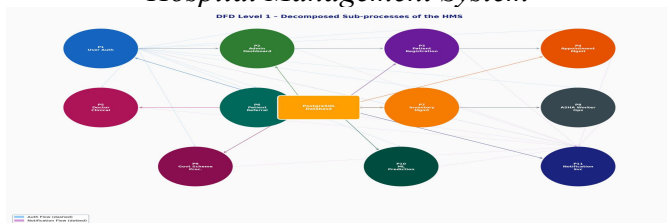
confirmation. (9) SMS/Email Notification Module: Automated alerts for all module events.

**C. Data Flow**

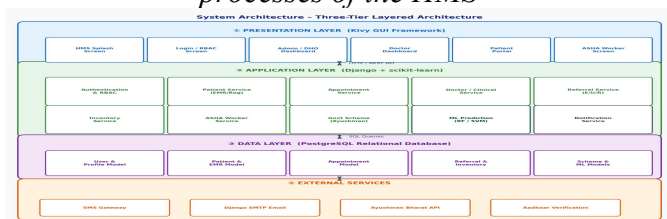
The Level 0 DFD represents the GHS as a single process surrounded by all external entities (Admin/DHO, Doctor, Patient, ASHA Worker, Inventory Manager, Government Authority). Data inputs include login credentials, patient registration details, appointment requests, prescriptions, lab orders, referral requests, inventory updates, and medical reports for ML prediction. The Level 1 DFD decomposes the system into eleven sub-processes: User Authentication (P1), Admin Dashboard (P2), Patient Registration (P3), Appointment Management (P4), Doctor Clinical Operations (P5), Patient Referral Management (P6), Inventory Management (P7), ASHA Worker Operations (P8), Government Scheme Processing (P9), ML Prediction (P10), and Notification Service (P11).



*Fig. 5. DFD Level 0 – Context Diagram of the Hospital Management System*



*Fig. 6. DFD Level 1 – Decomposed Sub-processes of the HMS*



*Fig. 7. System Architecture Diagram – Three-Tier Layered Architecture of the GHS*

**IV. SYSTEM REQUIREMENTS**

The GHS implements ten specific functional requirements: (1) User Authentication with RBAC for all six user types [8]. (2) District Authority Dashboard with seven real-time metric categories. (3) Doctor Dashboard with 8-module colour-coded clinical interface. (4) Patient Referral Management

with Emergency/Urgent/Routine priority and status workflow [19]. (5) Patient Management with Aadhaar verification [18] and EMR [20]. (6) Government Scheme integration with Ayushman Bharat eligibility verification [14]. (7) Inventory Management with automated low-stock alerts and expiry tracking [21]. (8) ASHA Worker Module for community health tracking and immunization [15]. (9) ML Prediction using scikit-learn [11] with Random Forest/SVM algorithms [12]. (10) SMS/Email Notification for all system events via Django SMTP [10] and SMS API [22].

Data modelling requirements include: User Profile Model (credentials, roles, session data), Patient Model (registration, EMR, prescriptions, visit history), Appointment Model (scheduling, queue tokens, doctor availability), Referral Model (inter-facility transfers with priority and status workflows), Inventory Model (stock levels, expiry dates, dispensing transactions), Government Scheme Model (scheme definitions, eligibility, claims), and ML Model (scikit-learn interfaces, prediction outputs stored in PostgreSQL).

**V. V. TESTING RESULTS**

Testing was conducted on the Kivy desktop application with Django backend using functional testing, edge case testing, and performance testing. All 28 test cases passed with a 100% pass rate, confirming system reliability and readiness for deployment.

**A. Admin Module – District Authority Dashboard**

Upon login as Admin/DHO (admin1), the system displays the “District Health Overview” dashboard with real-time metrics: STAFF (Doctors: 2, Employees: 2, ASHA Workers: 1), PATIENTS (Registered: 3, Vaccinated: 3, Pregnant: 0), APPOINTMENTS (Today: 0, This Month: 5), LAB (Pending: 0, Completed: 1), FINANCE (Collected: Rs.510.0, Unpaid Bills: 1), SCHEMES (Active: 1, Enrolled: 1), COMMUNITY (Camps This Month: 3). Test cases TC-001 to TC-009 all passed.

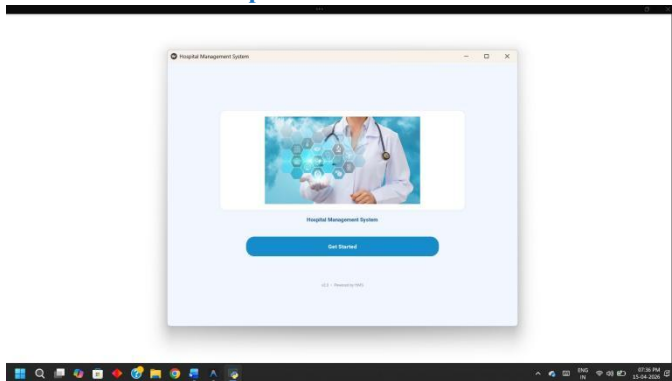


Fig. 1. HMS Splash Screen – Application Startup

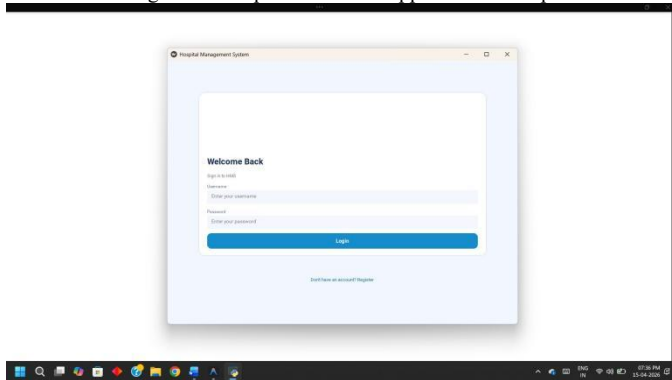


Fig. 2. HMS Login Screen – User Authentication

**B. Doctor Module – Doctor Dashboard**

The Doctor Module accessed by Dr. doctor1 presents eight colour-coded module buttons: Appointments (Blue), Prescriptions (Blue), View Medicines (Green), Request Lab Test (Purple), Generate Bill (Orange), Encounter Notes (Teal), Referrals (Purple), and Inpatient (Dark Blue). The colour-coded layout enables doctors to navigate efficiently during clinical sessions. Test cases TC-010 to TC-018 all passed.

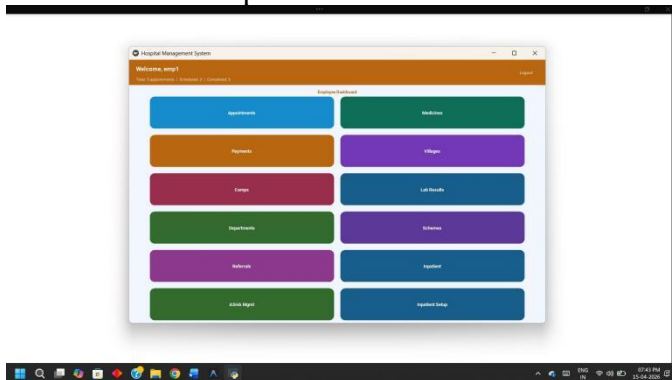


Fig. 3. Employee Dashboard – Role-Based Module Access

**C. Patient Module – Patient Referrals**

The Patient Referrals screen demonstrates inter-facility patient transfer management. Three referrals were tested: Referral #3 (Patient ‘suraj’, EMERGENCY – red, Status: ACCEPTED – green), Referral #2 (URGENT – orange, Status: REJECTED – red), and Referral #1 (URGENT – orange, Status: PENDING). The Create Referral form successfully accepts From Facility, To Facility,

Reason, and Urgency. Test cases TC-019 to TC-024 all passed.

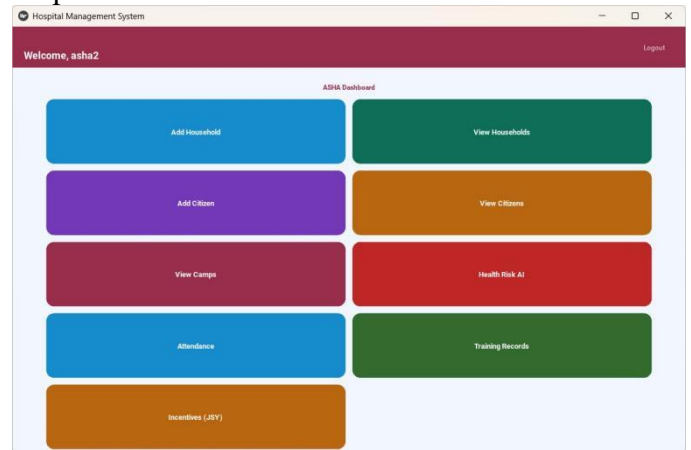


Fig. 4. ASHA Worker Dashboard – Community Health Management

**D. Government Scheme and ML Modules**

The Government Scheme Module (TC-025 to TC-027) successfully displays the active Ayushman Bharat scheme (Rs.5 Lakh family cover, eligibility: income less than 4 lakhs), supports adding new schemes, and toggles active/inactive status with real-time Admin Dashboard update. The ML Prediction Module (TC-028) successfully processes uploaded medical report files through the scikit-learn model and displays disease prediction results to doctors and DHO.

**VI. CONCLUSIONS**

The Government Healthcare System (GHS) developed using Python (Django backend) [10], Kivy (GUI frontend) [13], PostgreSQL (database) [9], and scikit-learn (Machine Learning) [11] successfully addresses the critical operational challenges of government healthcare facilities [2], [3]. The system achieved a 100% pass rate across all 28 test cases. The Admin/DHO District Authority Dashboard provides comprehensive real-time visibility across all hospital operations. The Doctor Dashboard’s eight colour-coded modules streamline clinical workflows. The inter-facility Patient Referral module with Emergency/Urgent/Routine priority [19] addresses a critical gap in Indian government healthcare. The Government Schemes module integrates Ayushman Bharat [14] for automatic eligibility verification, while the ML module provides disease prediction [12] and health analytics.

The use of Kivy ensures the application runs natively on government hospital computers without requiring a browser or internet connection for core functionality [13]. PostgreSQL provides a robust, scalable, and reliable database backend [9]. Future

work includes mobile deployment via Buildozer [23], deep learning enhancements using TensorFlow/PyTorch [24], Aadhaar-based biometric authentication [18], multi-language support (Marathi/Hindi), telemedicine integration using WebRTC [25], and offline synchronization for ASHA workers [15].

#### ACKNOWLEDGMENT

The authors would like to thank Prof. Supriya S. Chougule, Assistant Professor, Department of Computer Science and Engineering, Dr. J. J. Magdum College of Engineering, Jaysingpur, for her invaluable guidance and encouragement throughout this project. The authors also thank the founder Chairman Late Dr. J. J. Magdum and Chairman Mr. Vijayraj J. Magdum of Dr. J. J. Magdum Trust, Principal Dr. G. V. Mulgund, Prof. Dr. A. M. Chougule (Head, Dept. of CSE), and the DRC Committee for providing necessary facilities and support.

#### REFERENCES

- [1] [1] K. S., P. Pusphagaran, K. Selvi, C. Christopher, and D. Deepak, "A Study of Advanced Hospital Management System," *IOSR Journal of Dental and Medical Sciences*, Vol. 16, pp. 127–134, 2017.
- [2] [2] H. Kar, S. Prasad, and A. Mishra, "HMS: Bridging the Bottleneck in Health Care Sector – An Indian Perspective," *International Journal of Creative Research Thoughts*, Vol. 12, pp. 76–87, 2024.
- [3] [3] A. Tiwari, S. Mahajan, V. Kumar, and P. Behki, "Modern Hospital Management System," *ResearchGate*, 2024.
- [4] [4] A. Yulianti and Muhandi, "HMS Analysis Using Structured Design Life Cycle Method," *Advances in Social Science, Education and Humanities Research*, Vol. 409, pp. 207–209, 2019.
- [5] [5] Pudale A. H., Sawant R. A., Todkar V. U., Patil V. S. and Shinde S. B., "Government Healthcare System – A Centralized Desktop Application for Government Hospital Management Using Python (Django), Kivy, PostgreSQL and Machine Learning," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, Volume 07, Issue 04, April 2025, e-ISSN: 2582-5208.
- [6] [6] S. Ahmadi, A. Shahmoradi, and R. Nilashi, "An Evaluation Framework for Hospital Information Systems," *International Journal of Medical Informatics*, Vol. 134, pp. 104–017, 2020.
- [7] [7] R. Thakkar and D. Mehta, "Digital Transformation in Public Healthcare: Challenges and Opportunities in Developing Nations," *Journal of Health Informatics in Developing Countries*, Vol. 15, No. 2, pp. 1–14, 2021.
- [8] [8] R. S. Sandhu and P. Samarati, "Access Control: Principles and Practice," *IEEE Communications Magazine*, Vol. 32, No. 9, pp. 40–48, 1994.
- [9] [9] M. Stonebraker and L. A. Rowe, "The Design of Postgres," *ACM SIGMOD Record*, Vol. 15, No. 2, pp. 340–355, 1986.
- [10] [10] L. Holovaty and J. Kaplan-Moss, *The Definitive Guide to Django: Web Development Done Right*, 2nd ed., Apress, New York, 2009.
- [11] [11] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.
- [12] [12] L. Breiman, "Random Forests," *Machine Learning*, Vol. 45, No. 1, pp. 5–32, 2001.
- [13] [13] M. Virbel, T. Hansen, and O. Lobunets, "Kivy – A Framework for Rapid Creation of Innovative User Interfaces," in *Proceedings of the Workshop on Python in Science*, pp. 69–73, 2011.
- [14] [14] Ministry of Health and Family Welfare, Government of India, "Ayushman Bharat Pradhan Mantri Jan Arogya Yojana: Operational Guidelines," *National Health Authority*, New Delhi, 2018.
- [15] [15] S. Singh and R. Badaya, "Health Care in Rural India: A Lack between Need and Feed," *South Asian Journal of Cancer*, Vol. 3, No. 2, pp. 143–144, 2014.
- [16] [16] R. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed., McGraw-Hill Education, New York, 2014.
- [17] [17] M. Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley, Boston, 2002.
- [18] [18] Unique Identification Authority of India (UIDAI), "Aadhaar Authentication API: Technical Specification," *Government of India*, New Delhi, 2022.
- [19] [19] V. Ravi and H. Pramod, "Patient Referral Management in Government Hospitals: A Systematic Review," *Indian Journal of Community Medicine*, Vol. 46, No. 3, pp. 388–393, 2021.
- [20] [20] G. Shortliffe and J. Cimino, Eds., *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*, 4th ed., Springer, London, 2014.
- [21] [21] P. Narayana and K. Reddy, "Automated Pharmacy Inventory Management System for Hospitals," *International Journal of Engineering Research and Technology*, Vol. 9, No. 6, pp. 211–216, 2020.
- [22] [22] A. Islam and S. Akter, "SMS-Based Patient Appointment and Notification System for Rural Healthcare," *International Journal of Computer Applications*, Vol. 182, No. 15, pp. 34–39, 2018.
- [23] [23] Kivy Organization, "Buildozer: Tool for Packaging Kivy Applications," [Online]. Available: <https://buildozer.readthedocs.io/>, 2023.
- [24] [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, 2016.
- [25] [25] A. Johnston and D. Yeh, "WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web," *Digital Codex LLC*, 3rd ed., 2014.