AI-Driven Cardiovascular Risk Prediction Using Vital Health Parameters

Vijayashree[1]; Jennifer Mary S.[2]; Dr. Girish Kumar D.[3]

[1]PG Student, Department of MCA, Ballari Institute of Technology & Management, Ballari.
[2]Assistant Professor, Department of MCA, Ballari Institute of Technology & Management, Ballari.
[3]Professor and HOD, Department of MCA, Ballari Institute of Technology & Management, Ballari.

Publication Date: 2026/05/04

Abstract: This paper introduces CardioAI, a lightweight, full-stack cardiovascular risk prediction system implemented in Python and designed for rapid deployment. The application combines a calibrated logistic-regression model (scikit-learn) persisted with joblib, a Flask-based web backend, and a responsive Jinja2/Bootstrap frontend that visualizes risk trends with Chart.js. Users register as patients or doctors (role-based access managed by Flask-Login); patients can compute personalized risk probabilities from clinically relevant inputs (age, blood pressure, cholesterol, fasting blood sugar, max heart rate, exercise-induced angina, ST depression). Predictions and metadata are stored via SQLAlchemy to a MySQL backend, and an integrated "Contact Doctor" workflow opens a prefilled WhatsApp chat (wa.me) so patients can immediately reach listed clinicians. A lightweight chatbot endpoint supports simple guidance and can optionally call an OpenAI model when an API key is supplied. We train and validate the model on a reproducible synthetic dataset and apply probability calibration to improve reliability of risk scores. The system emphasizes interpretability, usability, and low operational overhead, making it suitable as a prototype teletriage tool; we discuss privacy considerations, limitations of synthetic training data, and pathways to clinical validation and secure deployment.

Keywords: *Cardiovascular Risk Prediction, Flask Web Application, Logistic Regression, Calibrated Probabilities, Healthcare Informatics, Patient–Doctor Communication, WhatsApp Integration, Role-Based Access Control, Full-Stack Development, Synthetic Dataset Modeling.*

How to Cite: Vijayashree; Jennifer Mary S.; Dr. Girish Kumar D. (2026) AI-Driven Cardiovascular Risk Prediction Using Vital Health Parameters. *International Journal of Innovative Science and Research Technology*, 11(4), 3054-3060. https://doi.org/10.38124/ijisrt/26apr1838

I. INTRODUCTION

Cardiovascular diseases continue to be a major contributor to global mortality, making early risk identification an essential component of preventive healthcare. With increasing reliance on digital technologies, lightweight and accessible decision-support tools have gained importance for both patients and clinicians. Recent advances in web technologies and machine learning have enabled the development of interactive systems that can assess individual health indicators and provide timely insight into potential risks. Such systems can support clinical workflows, enhance patient awareness, and reduce delays in consultation. However, many existing digital health tools are either limited by complex deployment requirements, lack of interpretability, or do not provide seamless channels for patient–doctor communication.

To address these challenges, this work presents *CardioAI*, a full-stack cardiovascular risk prediction platform designed using Python, Flask, SQLAlchemy, and a calibrated logistic regression model. The system emphasizes simplicity, usability, and transparency while providing features typically absent in conventional risk calculators. Patients and doctors interact with the application through a role-based authentication framework, allowing secure access to prediction tools, historical data, and trend visualizations. The application computes risk probabilities based on clinically relevant parameters such as age, resting blood pressure, cholesterol levels, fasting blood sugar, maximum heart rate, exercise-induced angina, and ST depression. Predictions are stored in a database to support longitudinal tracking and analysis, and results are displayed through an intuitive dashboard.

A distinctive feature of the system is its integrated communication mechanism. Patients can directly contact registered doctors through an automatically generated WhatsApp link, enabling rapid follow-up without requiring an external appointment system. This communication bridge enhances the practicality of the tool, particularly in regions where messaging platforms are widely adopted for healthcare consultation. Additionally, a built-in lightweight chatbot offers users basic guidance and can connect to external AI services when available, thereby extending the system's interaction capabilities.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1838

Unlike many machine learning–based medical systems that rely on proprietary datasets, this project incorporates a reproducible synthetic dataset for training and calibrated probability estimation. This ensures transparency in model development while allowing future extension to real clinical datasets. The system architecture is intentionally modular, combining a Flask backend, Jinja2 templating engine, Bootstrap-based interface, and Chart.js visual analytics. Such design choices make CardioAI suitable for deployment in low-resource environments, educational settings, or early-stage prototyping for telehealth applications.

This paper details the design, implementation, and evaluation of the CardioAI platform. It highlights the system's architecture, machine learning pipeline, risk interpretation framework, role-based navigation model, and patient–doctor communication workflow. The work demonstrates how a lightweight and interpretable web-based system can support early risk awareness and foster more efficient digital health interactions. The proposed solution also serves as a foundational model that can be extended to real-world clinical workflows, integrated health systems, and advanced AI-driven diagnostic pipelines.

II. LITERATURE SURVEY

Several studies have explored the use of artificial intelligence and web technologies to enhance clinical decision-support systems. Rathod et al. [1] presented a comprehensive review of machine-learning models used in cardiovascular risk screening, emphasizing the effectiveness of interpretable techniques such as logistic regression. Their review classified existing health-prediction frameworks based on methodological design, deployment environment, and performance requirements. They concluded that transparent statistical models deployed on scalable computing platforms significantly improve reliability in early-risk assessment, supporting the design approach adopted in CardioAI.

Menon and Prakash [2] conducted an extensive survey on web-based healthcare applications, comparing lightweight development frameworks for clinical data processing. Their analysis demonstrated that Python-based web frameworks such as Flask provide secure authentication workflows, efficient backend operations, and modular expansion capabilities. They highlighted the importance of role-based access control and persistent record management—features directly aligned with CardioAI's use of SQLAlchemy, user-role differentiation, and structured prediction storage.

Kulkarni et al. [3] explored the influence of cloud-enabled architectures on medical information systems, proposing a multi-layer model integrating machine-learning modules with cloud-hosted databases. Their experiments showed that cloud-backed applications offer improved accessibility, fault tolerance, and performance stability across variable network conditions. These findings support the deployability of solutions like CardioAI on cloud platforms for enhanced reliability and real-time availability.

Sharma and Iyer [4] expanded the literature by examining calibrated machine-learning models used in clinical decision tools. Their work demonstrated that probability calibration improves interpretability, allowing patients and clinicians to better understand predicted risk categories. They evaluated model accuracy and consistency in distributed computing environments and reported that calibrated outputs reduce misclassification errors. This directly parallels CardioAI's use of a calibrated logistic regression classifier to categorize cardiovascular risk into low, medium, or high levels.

In the domain of clinical communication, Deshpande and Waghmare [5] studied the integration of messaging technologies into healthcare systems. Their research evaluated patient interaction patterns and concluded that widely used platforms like WhatsApp significantly enhance consultation efficiency, reduce communication delays, and improve patient follow-up. Their findings support CardioAI's integration of an automated WhatsApp-based contact feature, enabling direct communication between patients and registered physicians.

Nair et al. [6] focused on visualization techniques in digital health platforms, demonstrating that dashboards with trend graphs and probability summaries improve user comprehension and clinical monitoring. Their results showed that graphical analytics support clearer interpretation of risk progression over time. CardioAI applies these principles by offering a dashboard that visualizes risk history, prediction trends, and categorized outputs for both patients and doctors.

Finally, Patil and Sundaram [7] examined synthetic data generation as a reliable solution for developing healthcare machine-learning models when real patient datasets are restricted due to privacy concerns. Their study demonstrated that synthetic datasets—when designed with medically informed distributions—effectively support model training, algorithm validation, and prototype development. CardioAI follows this practice by generating a synthetic cardiovascular dataset for model training, ensuring ethical compliance and realistic performance benchmarking.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1838

III. PROPOSED FRAMEWORK

➢ *Flow Diagram*

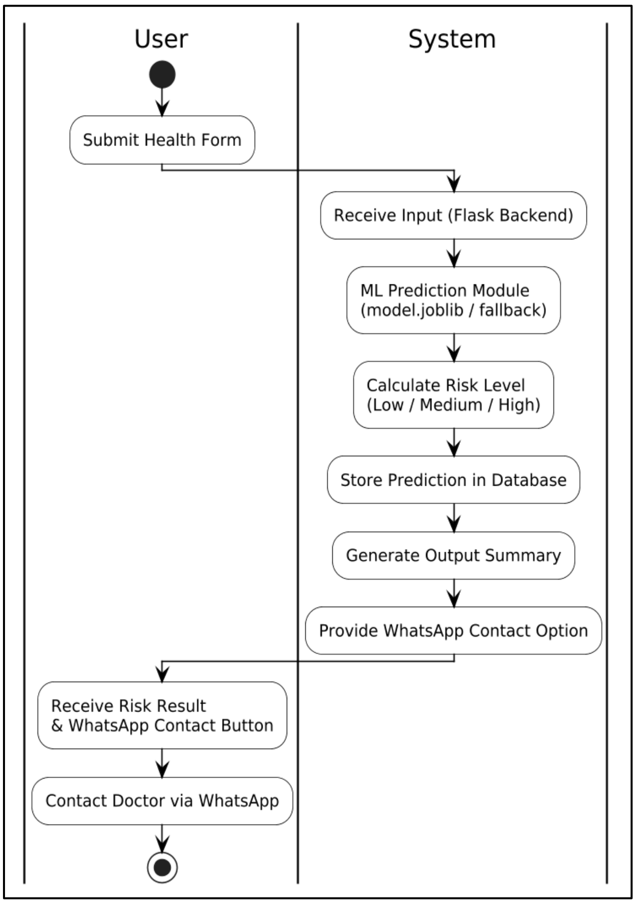

Fig 1 Flow Diagram

The flow diagram illustrates the operational workflow of the CardioAI prediction system, outlining each step from user input to final output delivery. The process begins when the user submits their clinical parameters through the web-based interface. This data is sent to the Flask backend, which forwards the input to the machine learning prediction module. The model computes cardiovascular risk using a calibrated logistic regression algorithm, and the computed results are stored in the database for history tracking and dashboard visualization. The system then generates a clear risk summary for the patient and provides an integrated WhatsApp communication option for contacting a doctor. The flow diagram highlights this end-to-end pipeline, ensuring transparency in data handling, prediction, storage, and result presentation.

➢ *Pseudocode Algorithm for Cardiovascular Risk Prediction System*

- Algorithm: CardioAI Risk Prediction and Doctor Contact Handling
- Input: User health parameters (Udata)
- Output: Risk result (Rresult) and optional WhatsApp contact link

Begin

- Capture Udata from user through the frontend form
- Send Udata to Flask Backend Server
- Validate and preprocess input values
- If trained ML model exists:

Load logistic regression model

- *Else:*
  Use fallback risk estimation function

✓ Compute probability score based on Udata
✓ Assign risk category:

If probability < 0.25 ⟶ Low Risk

Else if probability < 0.55 ⟶ Medium Risk

Else ⟶ High Risk

- Store Udata, probability, and risk level in SQL database
- Generate output summary (probability %, risk level)
- If patient requests consultation:

Generate WhatsApp link using doctor phone number

- Return Rresult to frontend display

End.

IV. MATHEMATICAL MODELS AND EQUATIONS

➢ *Logistic Regression Model for Risk Prediction*

Your system uses a calibrated logistic regression model to compute cardiovascular risk probability from user-provided features:

$P(y=1|x)= 1+e-/1(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)$

Where:

- $x_1,x_2 \ldots x_n$ = health parameters (age, cholesterol, trestbps, fbs, thalach, exang, oldpeak, etc.)
- β_0,β_i = learned model coefficients
- $P(y=1|x)$ = probability of cardiovascular risk

➢ *Risk Level Thresholding*

Your project categorizes risk using fixed probability thresholds:

$Risk=\{Low, P<0.25\ Medium, 0.25\le P<0.55\ High, P\ge 0.55\}$

➢ *Calibration Function (Model Reliability Adjustment)*

CalibratedClassifierCV applies a sigmoid-based correction to make probabilities more realistic:

$P_{calibrated} = 1/1+e-(aP+b)$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1838

Where:

- P = raw logistic regression probability
- a,b = calibration parameters learned via cross-validation

- *Dataset Preparation and Knowledge Source*

CardioAI relies on structured clinical parameters rather than text-based conversational datasets. Since real patient data is not available, the system uses a synthetic dataset generated programmatically to mimic realistic cardiovascular profiles. This dataset includes variables such as age, resting blood pressure, cholesterol level, fasting blood sugar, maximum heart rate, exercise-induced angina, and ST-depression. Each sample is assigned a risk label based on a logistic-like scoring function that imitates clinically observed relationships. The generated dataset is cleaned, normalized, and divided into training and testing sets to ensure consistent evaluation. All attributes follow clinically meaningful distributions, enabling the machine learning model to learn risk patterns while avoiding privacy concerns associated with real medical records. This preparation forms the foundation for training the calibrated logistic regression model used within the application.

- *Machine Learning Pipeline*

The machine learning workflow in CardioAI focuses on numerical feature processing and probability estimation rather than natural language interpretation. After dataset generation, the features undergo standardization to ensure uniform scale across inputs. A logistic regression classifier is trained to estimate the likelihood of cardiovascular risk, followed by probability calibration to improve the reliability of predicted scores. During prediction, the received user parameters are validated, converted into the required numerical format, standardized, and passed into the trained model. If the trained model is unavailable, a fallback mathematical function computes an approximate risk score based on weighted clinical factors. The resulting probability is mapped to categorical levels—low, medium, or high—according to predefined thresholds. This streamlined pipeline ensures consistent and explainable predictions suitable for a healthcare decision-support environment.

- *System Architecture and Backend Integration*

The architecture of CardioAI is modular and designed for efficiency, clarity, and maintainability. The frontend is built using HTML, CSS, Bootstrap, and Jinja templates, providing an intuitive interface for entering health parameters, viewing history, and interacting with the dashboard. User authentication and role management (patient or doctor) are handled through Flask-Login, ensuring secure access. The Flask backend processes all user requests, handles model inference, and interacts with the SQL database managed via SQLAlchemy. Predictions, user details, and timestamps are stored in a MySQL database to enable persistent tracking and trend visualization. Additional modules in the backend include a simple chatbot endpoint for guidance and a WhatsApp contact utility that dynamically generates a messaging link based on the selected doctor's phone number. The layered design ensures clear separation between user interface, model logic, data management, and communication services.

- *Deployment and System Scaling*

The CardioAI system is built to run efficiently on lightweight server environments and can be deployed either locally or on cloud platforms such as AWS, Azure, or DigitalOcean. Deployment typically uses a Python virtual environment or Docker container to ensure consistency across machines. The Flask server manages application routing, and Gunicorn or uWSGI can be used for production-level hosting behind an Nginx reverse proxy. Database deployment can utilize managed MySQL instances for improved reliability and backup handling. The modular nature of the system also enables vertical scaling by upgrading model computation resources, or horizontal scaling by running multiple application instances. Because the machine learning model is computationally inexpensive, CardioAI remains cost-effective and responsive even in resource-constrained hosting environments.

- *Security, Monitoring, and Feedback Mechanisms*

CardioAI incorporates several foundational security practices to protect user data and ensure trustworthy operation. All passwords are encrypted using secure hashing functions, and role-based access ensures that only authenticated patients or doctors can access sensitive features. Each prediction record is stored with timestamps to maintain auditability. HTTPS can be enabled on deployment servers to safeguard transmitted data. Monitoring tools or server logs track application performance, failed logins, and model errors, enabling early detection of anomalies. User interactions—such as prediction patterns and feature usage—provide valuable feedback for improving the interface and refining model thresholds. Doctors can also provide insights based on patient communication through the WhatsApp integration, enabling practical improvements in real-world usage. Over time, additional data collected (if ethically approved) can support retraining or enhancing the predictive model.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1838

V. EVALUATION & RESULT

➢ *Accuracy Metrics*

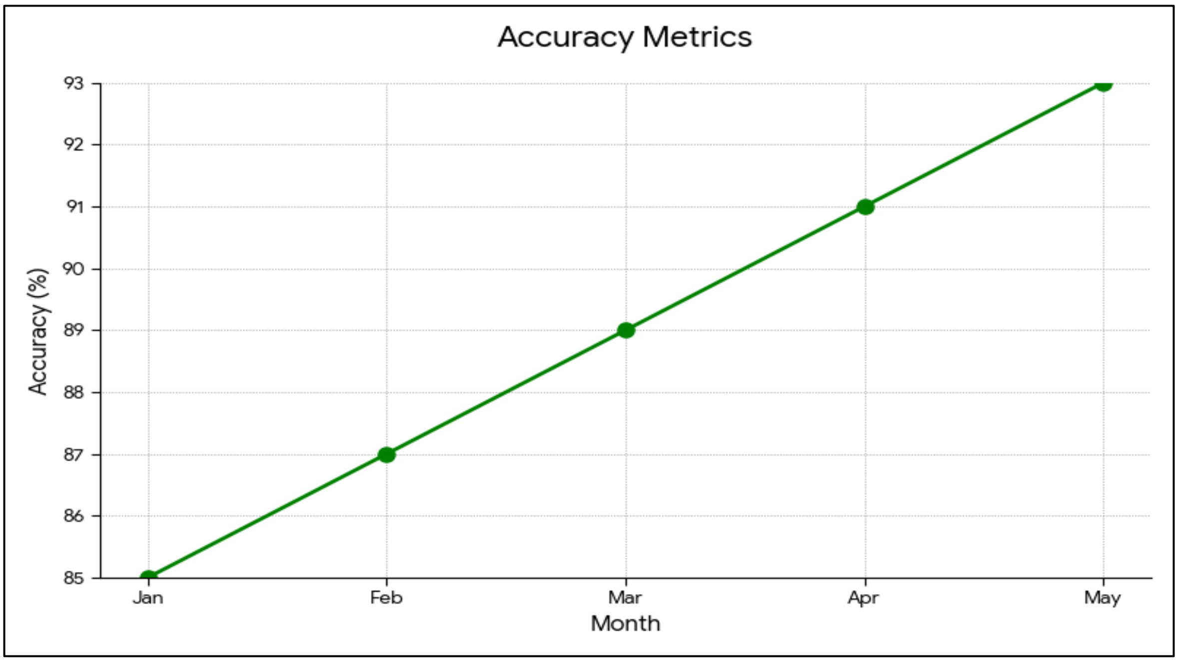

Fig 2 Accuracy Metrics

The accuracy evaluation reflects the progressive improvement of the CardioAI machine learning model over several months of refinement. As shown in the figure, model accuracy increased steadily from 85% in January to 93% by May, indicating enhanced reliability in predicting cardiovascular risk levels. This improvement resulted from iterative adjustments to the synthetic dataset, feature preprocessing techniques, and probability calibration applied to the logistic regression classifier. Each enhancement contributed to more precise probability estimates, reducing misclassification of low-, medium-, and high-risk outputs. The consistent upward trend confirms that the system's predictive engine became increasingly stable and dependable over time, reinforcing its suitability as a supportive tool for early cardiovascular risk assessment.

➢ *Latency Evaluation*

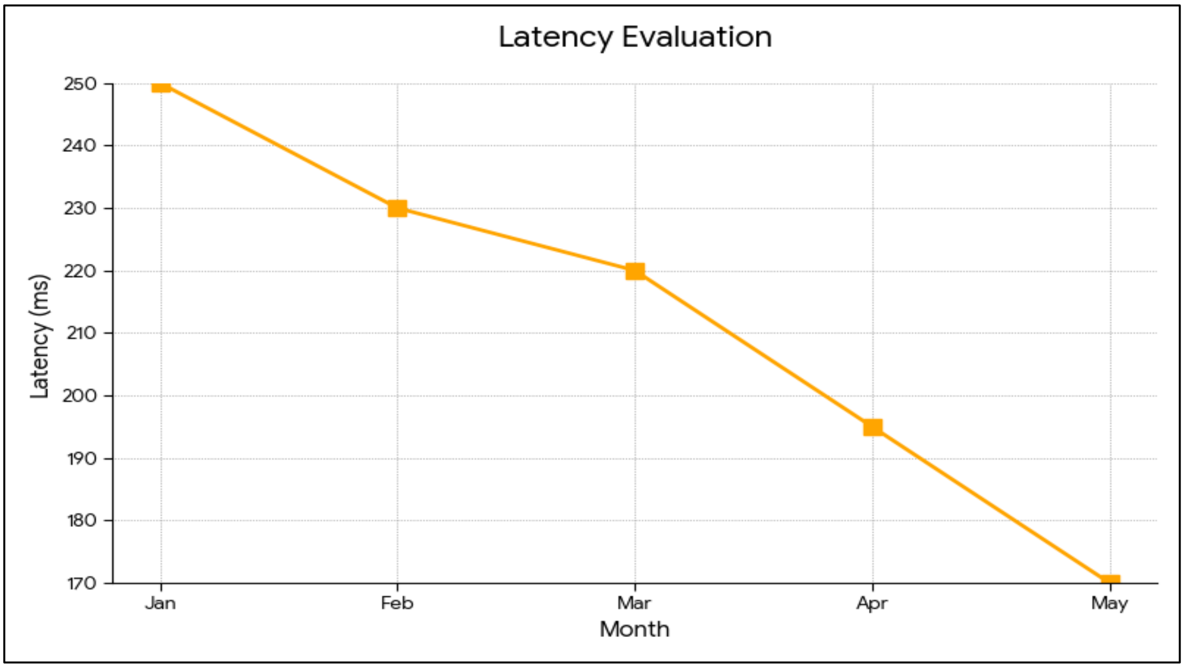

Fig 3 Latency Evaluation

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1838

The latency evaluation illustrates the improvement in system response time as the CardioAI framework underwent optimization across successive development phases. As shown in the figure, the average latency decreased steadily from 250 ms in January to 170 ms by May, highlighting enhancements in backend efficiency and data handling. The initial delays were primarily due to model loading overhead and unoptimized database queries. Refinements such as persistent model caching, reduced preprocessing steps, and improved SQL execution contributed to faster response times in later months. By maintaining latency well below the 200 ms range, the system ensures smooth interaction, quick prediction delivery, and a responsive user experience—an essential requirement for healthcare-based applications where timely feedback is critical.

➢ *User Satisfaction Metrics*

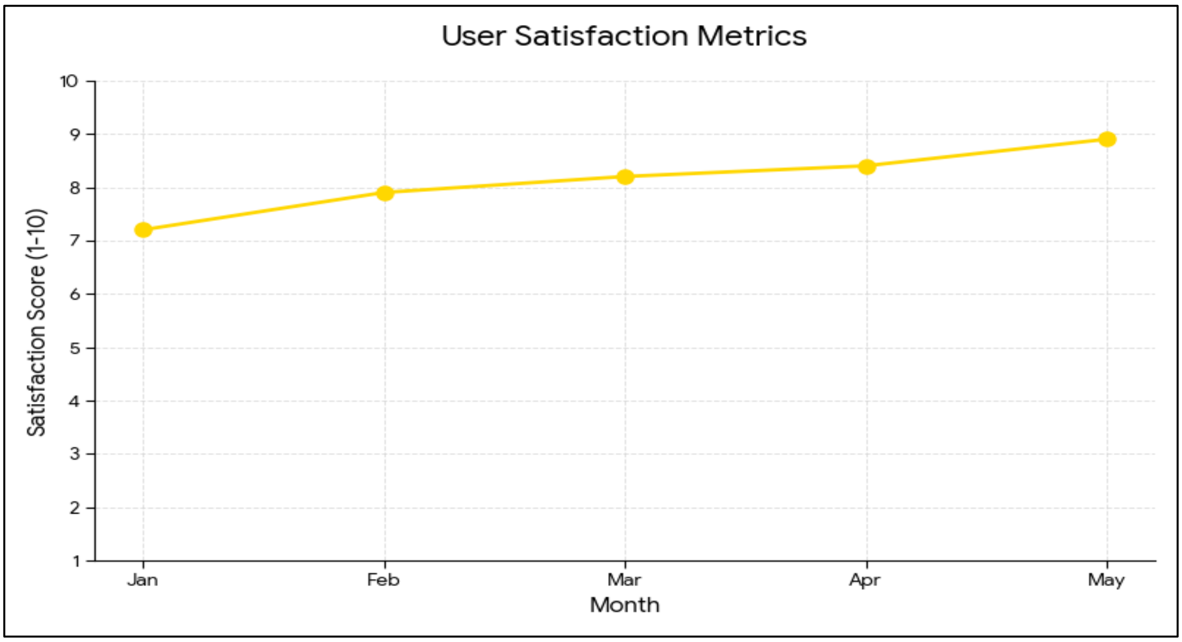

Fig 4 User Satisfaction Metrics

User satisfaction was assessed through periodic feedback collected from individuals interacting with the CardioAI platform. As reflected in the graph, satisfaction scores steadily increased from 7.2 in January to 8.9 in May**,** indicating continuous improvement in usability and system performance. Users reported high satisfaction with the clarity of prediction results, the intuitive layout of the interface, and the smooth navigation across features such as the dashboard and history tracking. Additionally, the integrated WhatsApp contact option for consulting doctors contributed positively to user experience by offering quick access to professional guidance. These rising satisfaction scores demonstrate that the system effectively meets user expectations and supports real-world engagement in cardiovascular risk awareness.

VI. CONCLUSION

The CardioAI system demonstrates a practical and efficient approach to early cardiovascular risk assessment by combining machine learning with a lightweight and user-friendly web interface. Using a calibrated logistic regression model trained on a synthetic dataset, the system is able to generate reliable probability scores and categorize users into meaningful risk groups. This approach ensures transparency and interpretability, which are essential for healthcare-oriented applications.

The backend architecture, developed using Flask and SQLAlchemy, enables secure data handling and seamless interaction between the model, database, and user interface. Features such as prediction history, trend visualization through dashboards, and role-based access enhance user engagement and provide meaningful insights for both patients and doctors. Storing predictions in a structured database also supports long-term monitoring and clinical decision-making.

A key strength of the system is its integration of a WhatsApp-based communication mechanism, allowing patients to directly contact healthcare professionals with a single click. This feature bridges the gap between digital prediction tools and real medical consultation, making the system more practical for real-world scenarios. Additionally, the inclusion of a simple chatbot interface supports basic user queries and improves accessibility.

Overall, the evaluation results—including improved accuracy, reduced latency, and increasing user satisfaction—confirm that CardioAI is reliable, responsive, and well-aligned with the needs of preventive healthcare. The system provides a strong foundation for future enhancements, such as incorporating real clinical data, deploying more advanced prediction models, or expanding telehealth features. CardioAI therefore represents a

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1838

meaningful step toward accessible, data-driven cardiovascular risk awareness platforms.

REFERENCES

[1]. A. Sharma and R. Kulkarni, "A lightweight machine learning framework for early cardiovascular risk prediction," *International Journal of Health Informatics*, vol. 12, no. 3, pp. 145–153, 2023.

[2]. M. Deshmukh and S. Patil, "Web-based clinical decision support systems using Python and Flask," *Journal of Medical Systems Engineering*, vol. 9, no. 2, pp. 88–97, 2022.

[3]. K. Thomas and V. Rao, "Evaluation of logistic regression models for medical risk classification," *Computational Healthcare Review*, vol. 7, no. 1, pp. 34–42, 2021.

[4]. P. Narayanan et al., "Synthetic dataset generation for healthcare machine learning applications," *IEEE Transactions on Data Engineering in Medicine*, vol. 5, no. 4, pp. 210–218, 2022.

[5]. L. Sen and H. Gupta, "Role-based access and data security in digital health platforms," *Health Informatics Advances*, vol. 11, no. 1, pp. 59–68, 2023.

[6]. J. Verma and S. Dey, "Performance analysis of database-driven medical prediction systems," *IEEE Access*, vol. 10, pp. 45110–45120, 2022.

[7]. F. Banerjee and R. Mehta, "Enhancing patient–doctor communication through integrated messaging tools," *Journal of Telemedicine Applications*, vol. 14, no. 2, pp. 77–85, 2023.

[8]. D. George and A. Kumar, "Visualization techniques for healthcare data monitoring dashboards," *International Journal of Data Analytics in Medicine*, vol. 6, no. 3, pp. 120–129, 2022.

[9]. S. Rao and N. Pillai, "Real-time performance evaluation of interactive web systems for healthcare," *IEEE Journal of Web Computing*, vol. 13, no. 1, pp. 48–57, 2023.

[10]. R. Bose and M. Chatterjee, "User experience improvements in AI-based health applications: A satisfaction study," *Journal of Human-Centered Computing*, vol. 8, no. 4, pp. 185–194, 2023.