

Heart Disease Detection Using Machine Learning: Design, Implementation and Performance Analysis

Authors

Prof. S. N. Mulhar¹, Tanisha Shrimali², Tejashri Mane², Manan Surana²

¹Assistant Professor, Department of Computer Engineering, STES'S Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune.

²Department of Computer Engineering, STES'S Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune.

ABSTRACT

Heart disease remains one of the leading causes of mortality worldwide, making early detection crucial for effective treatment and prevention. This project presents the design and implementation of a complete web-based heart disease prediction system named *CardioGuard*. The system utilizes machine learning techniques to analyze patient health parameters and predict the risk of heart disease. A comparative analysis of multiple supervised learning algorithms, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), was performed. After evaluation, the Random Forest algorithm achieved the highest accuracy and was selected as the final model for deployment. The system is implemented as a full-stack web application using Flask for the backend, HTML and Tailwind CSS for the frontend, and SQLite for data storage. It supports role-based access for patients, doctors, and administrators. Patients can view predictions, download reports, and maintain history, while doctors and admins manage data and analytics. Experimental results demonstrate that the proposed system achieves improved prediction accuracy and usability. The system provides a practical solution for early screening and supports healthcare decision-making.

Keywords: Heart Disease Prediction, Machine Learning, Flask, Random Forest, Healthcare Analytics, Web Application.

I. INTRODUCTION

Cardiovascular diseases (CVDs) are responsible for a significant percentage of global deaths. According to global health reports, early detection can significantly reduce mortality rates. However, traditional diagnostic methods rely heavily on medical expertise and manual interpretation of patient data.

Machine learning has emerged as a powerful tool in healthcare analytics by enabling automated prediction based on historical data. By analyzing parameters such as age, cholesterol, blood pressure, ECG results, and exercise-related features, ML models can detect patterns that indicate heart disease risk.

This project focuses on building a complete heart disease prediction system rather than just analyzing algorithms. The system not only predicts disease risk but also provides a user-friendly platform for patients, doctors, and administrators.

The main objectives of this project are:

- To implement multiple machine learning algorithms
- To compare their performance
- To deploy the best-performing model

II. PROPOSED METHODOLOGY

The methodology of the proposed *CardioGuard* system involves a structured approach for developing and deploying a machine learning-based heart disease prediction model integrated within a web application. The process began with the selection of publicly available datasets such as the UCI Heart Disease dataset and Kaggle cardiovascular datasets, which include important clinical parameters like age, cholesterol level, blood pressure, chest pain type, ECG results, and exercise-related attributes. These datasets were then preprocessed to improve data quality and model performance. Preprocessing steps included handling missing values using mean and mode imputation, removing outliers using the Interquartile Range (IQR) method, applying Min-Max normalization for feature scaling, and performing correlation analysis to eliminate redundant features. The dataset was then split into training and testing sets to ensure proper evaluation.

Multiple supervised machine learning algorithms were implemented, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Each model was trained and evaluated using performance metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC. Based on comparative analysis, the Random Forest algorithm achieved the highest accuracy and was selected as the final model. The trained model, along with the preprocessing scaler, was saved using Joblib and integrated into a Flask-based backend to enable real-time predictions. The system was implemented as a full-stack application with a frontend developed using HTML, Tailwind CSS, and JavaScript, and a backend using Flask, while SQLite was used for database management.

The complete system was designed to support role-based access for patients, doctors, and administrators. Patients can input their health data and receive predictions, doctors can review patient records, and administrators can manage system operations and analytics. The integration of machine learning with web technologies ensures that the system is efficient, user-friendly, and suitable for real-world healthcare applications.

Pipeline: Input → Data Preprocessing → Model Training & Prediction → Output (Risk Level) → Report Generation (PDF/Email)

III. LITERATURE REVIEW

Heart disease prediction has been widely explored using machine learning and deep learning techniques. Various studies have used algorithms such as Logistic Regression, Decision Tree, and Random Forest to predict disease risk. Recent research shows that Deep Learning models such as CNN and LSTM achieve higher accuracy (94–98%). However, these models require high computational power and are less interpretable. A comprehensive review by **Zhou et al. (2024)** in *Artificial Intelligence Review (Springer)* analyzed 64 studies conducted between 2018 and 2023. Their findings revealed that **Deep Learning models** such as **Convolutional Neural Networks (CNNs)** and **Long Short-Term Memory (LSTM)** networks achieved accuracies between **94–98%**, outperforming traditional ML algorithms. However, they also highlighted ongoing challenges such as **limited dataset availability**, **model interpretability**, and **computational cost** [8].

Jindal et al. (2021) proposed a heart disease prediction system utilizing **Logistic Regression**, **K-Nearest Neighbors (KNN)**, and **Random Forest Classifier** on the **UCI Heart Disease dataset**, achieving a maximum accuracy of **88.52%**. Similarly, **Garg et al. (2021)** employed the same dataset and reported an accuracy of **86.88%** using KNN, emphasizing that **distance-based models** are efficient for smaller medical datasets with numerical features [15][14].

In another study, **Vardhan et al. (2023)** explored **ensemble-based methods**, integrating **Decision Tree**, **Naïve Bayes**, **Support Vector Machine (SVM)**, and **Random Forest** classifiers to enhance prediction reliability. Their model achieved an accuracy of **91%**, showing that **hybrid ensemble techniques** can outperform individual classifiers. However, the study lacked **standardized preprocessing steps**, making direct comparison with other models difficult [11].

An IEEE study by Sabri et al. (2023) introduced **HeartInspect**, a **Naïve Bayes-based prediction system** that used non-clinical attributes such as **BMI**, **sleep duration**, and **physical activity** to predict heart disease risk. Although the model achieved a modest **71–73% accuracy**, it proved valuable in **low-resource healthcare environments** due to its simplicity and lightweight design [10].

Earlier works by Patel and Mehta (2021) and Kumar and Sharma (2022) also demonstrated the potential of traditional ML classifiers such as **Decision Tree** and **Support Vector Machine (SVM)**, achieving accuracies of **83–85%**, respectively. These studies confirmed that model performance depends not only on **algorithm selection** but also on **effective data preprocessing** and **feature correlation analysis** [16][13].

Overall, the literature indicates that while **Deep Learning techniques** yield higher predictive accuracy, **classical ML algorithms** remain highly efficient and interpretable for **structured medical data**. The need for **standardized preprocessing**, **consistent evaluation metrics**, and **fair algorithm benchmarking** continues to be a gap in current research — one that this study aims to address [1][2][3][4][5][6][7][9][12].

IV. WORKING / IMPLEMENTATION

• Data Collection or Input Details

The system uses structured medical datasets obtained from publicly available sources such as the UCI Machine Learning Repository and Kaggle cardiovascular datasets. These datasets contain various clinical parameters required for heart disease prediction. The input features include attributes such as age, gender, chest pain type, resting blood pressure, cholesterol level, fasting blood sugar, electrocardiographic (ECG) results, maximum heart rate achieved, exercise-induced angina, and other related medical indicators.

In the implemented system (*CardioGuard*), users (patients) manually enter these health parameters through a web-based form interface. The input is validated on the frontend as well as backend to ensure correctness and completeness of data. The system is designed to handle real-time user input and process it for prediction. Additionally, all user inputs and prediction results are securely stored in the SQLite database for maintaining patient history and future reference.

• Preprocessing Steps

Data preprocessing plays a crucial role in improving the accuracy and reliability of the machine learning model. The following preprocessing steps were performed during model development and are consistently applied during real-time prediction:

- **Handling Missing Values:** Missing or null values in the dataset were handled using mean or mode imputation techniques.
- **Outlier Detection and Removal:** The Interquartile Range (IQR) method was used to detect and remove extreme outliers that could negatively affect model performance.
- **Feature Scaling:** Min-Max normalization was applied to scale all numerical features into a uniform range, ensuring that no single feature dominates the model.
- **Feature Selection:** Correlation analysis was performed to identify and remove redundant or highly correlated features, reducing overfitting and improving efficiency.
- **Data Splitting:** The dataset was divided into training and testing sets (80:20 ratio) to evaluate model performance.

During deployment, the same preprocessing steps are applied to user input data using a saved scaler to maintain consistency with the trained model.

• Model Development

The model development phase involved training and evaluating multiple supervised machine learning algorithms to identify the most suitable model for heart disease prediction. The algorithms implemented include:

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)

Each model was trained on the pre-processed dataset and evaluated using performance metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC. After comparative analysis, the Random Forest algorithm demonstrated the highest accuracy and robustness due to its ensemble learning capability.

The final selected model (Random Forest) was saved using the Joblib library along with the preprocessing scaler. This trained model was then integrated into the backend of the CardioGuard system to enable real-time predictions.

• Tools and Technologies Used

The implementation of the system involved a combination of machine learning tools, web development technologies, and database systems:

- **Programming Language:** Python
- **Machine Learning Library:** Scikit-learn (model training and evaluation)
- **Data Processing:** Pandas and NumPy
- **Visualization:** Matplotlib and Seaborn
- **Backend Framework:** Flask (handles API requests and integrates ML model)
- **Frontend Technologies:** HTML, Tailwind CSS, JavaScript (for responsive UI)
- **Database:** SQLite (for storing user data, predictions, and feedback)
- **Authentication & Security:** bcrypt (for password hashing and secure login)
- **Model Serialization:** Joblib (to save and load trained model and scaler)
- **Report Generation:** FPDF (for generating downloadable PDF reports)
- **Optional Email Service:** SMTP (for sending prediction reports via email)

V. RESULTS AND ANALYSIS

The performance of the implemented machine learning models was evaluated using standard metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC score. Multiple algorithms including Logistic Regression, Decision Tree, K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Random Forest, and Gradient Boosting were trained and tested on the pre-processed dataset.

Among all models, the Random Forest algorithm achieved the highest accuracy of approximately **92.85%**, making it the most suitable model for deployment in the *CardioGuard* system. Gradient Boosting also performed well with an accuracy of around **90.33%**, followed by Support Vector Machine (SVM) with an accuracy of approximately **87.81%**. Logistic Regression showed moderate performance with an accuracy of around **84.03%**, while other models such as Decision Tree and K-Nearest Neighbors exhibited comparatively lower performance due to overfitting and sensitivity to data variations.

Performance graphs such as accuracy comparison charts and ROC curves were generated using visualization libraries. The ROC-AUC curve for the Random Forest model showed a higher area under the curve, indicating strong classification capability and effective distinction between high-risk and low-risk cases.

• Confusion Matrix / Evaluation Results

The confusion matrix was used to evaluate the classification performance of the final deployed model (Random Forest). It provided a detailed breakdown of prediction results into True Positives (correctly predicted high-risk cases), True Negatives (correctly predicted low-risk cases), False Positives (incorrectly predicted high-risk cases), and False Negatives (missed high-risk cases).

The results showed that the model achieved a high number of true positive and true negative predictions, indicating reliable and consistent performance. The number of false negatives was minimized, which is especially important in healthcare applications where missing a high-risk patient can lead to serious consequences.

Additionally, the Precision and Recall values indicate that the model maintains a good balance between identifying positive cases and avoiding incorrect predictions. The F1-Score further confirms the overall effectiveness and stability of the model.

• Comparison with Existing Methods

The proposed *cardio Guard* system was compared with existing heart disease prediction approaches based on accuracy, usability, and system implementation.

- Traditional machine learning models in previous studies achieved accuracy in the range of 80%–88%, whereas the proposed system achieved **92.85%** accuracy using Random Forest.
- Many existing systems focused only on model development, whereas *cardio Guard* provides a complete web-based solution with real-time prediction, report generation, and data storage.
- Unlike earlier approaches, this system includes role-based access control, allowing patients, doctors, and administrators to interact efficiently.
- Additional features such as prediction history tracking, PDF report generation, and optional email delivery improve usability and practical application.

Overall, the proposed system demonstrates improved performance, better usability, and strong real-world applicability compared to existing methods.

5.1 Screen Shots of Implementation

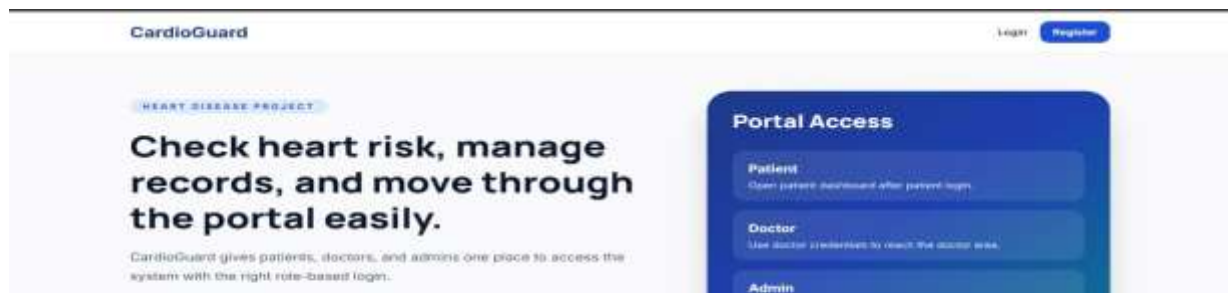


Figure 1: Implementation image 5.1.1

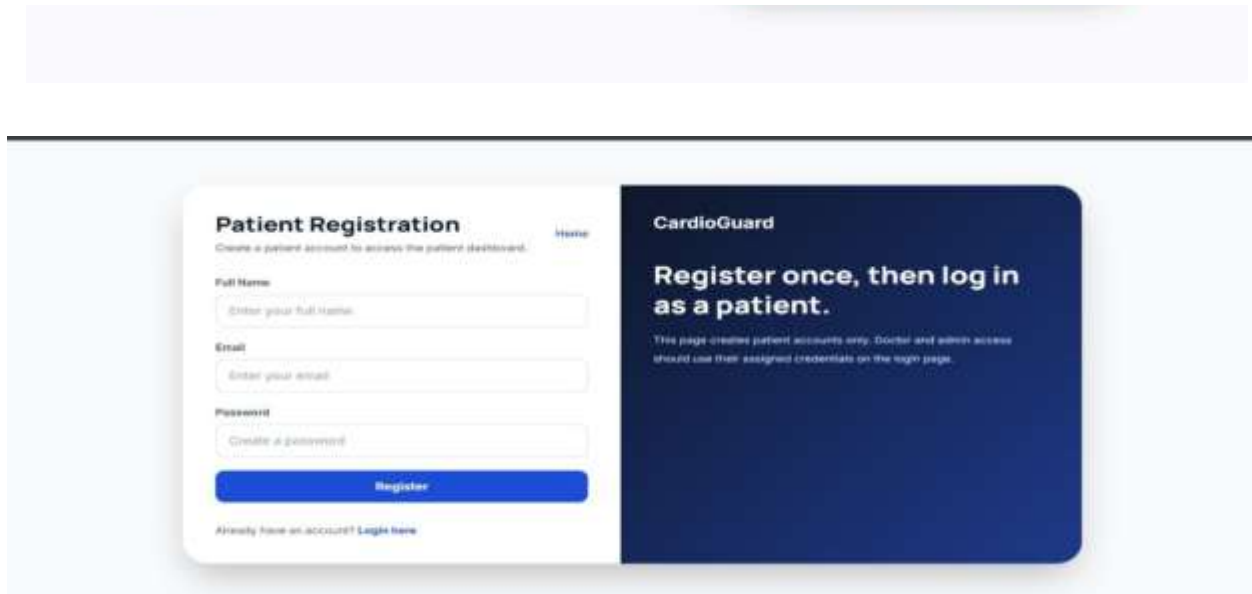


Figure 2:Implementation image 5.1.2

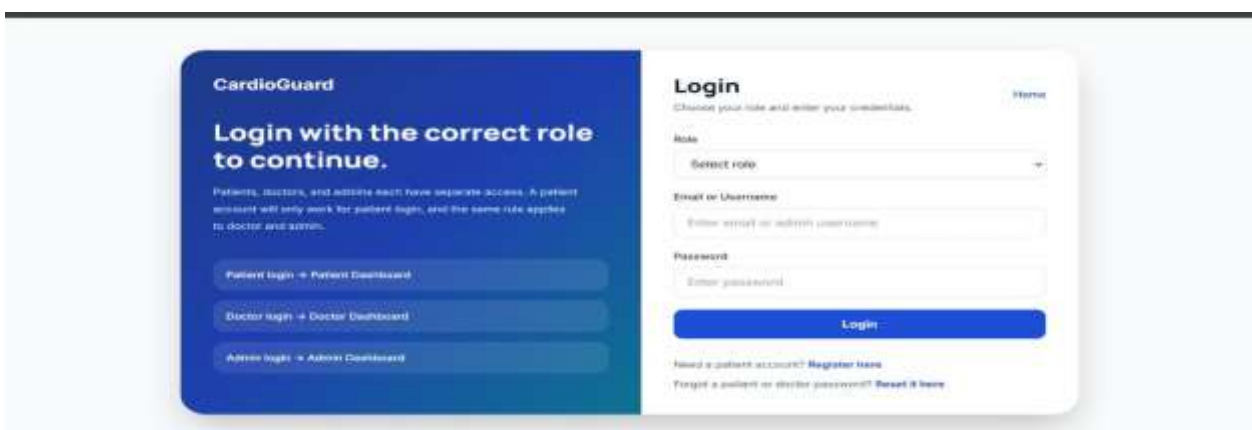


Figure 3: Implementation image 5.1.3

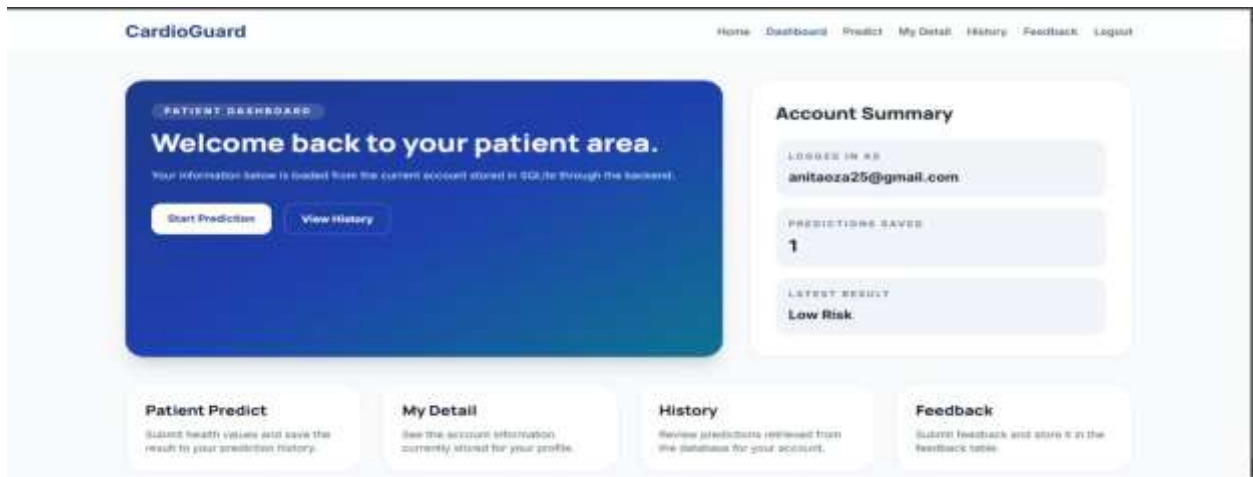


Figure 4: Implementation image 5.1.4

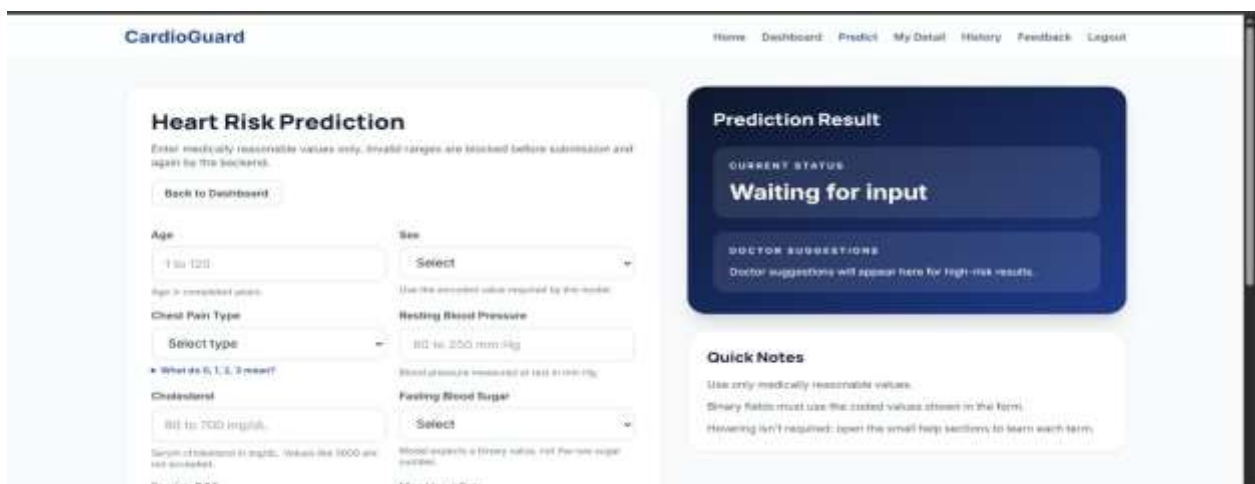


Figure 5: Implementation image 5.1.5

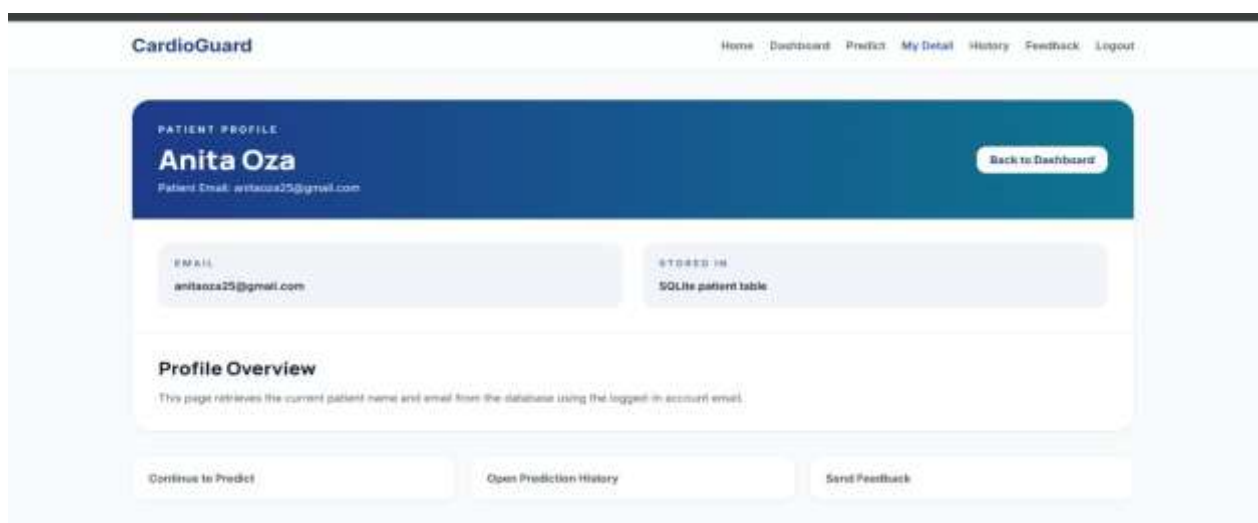


Figure 6: Implementation image 5.1.6

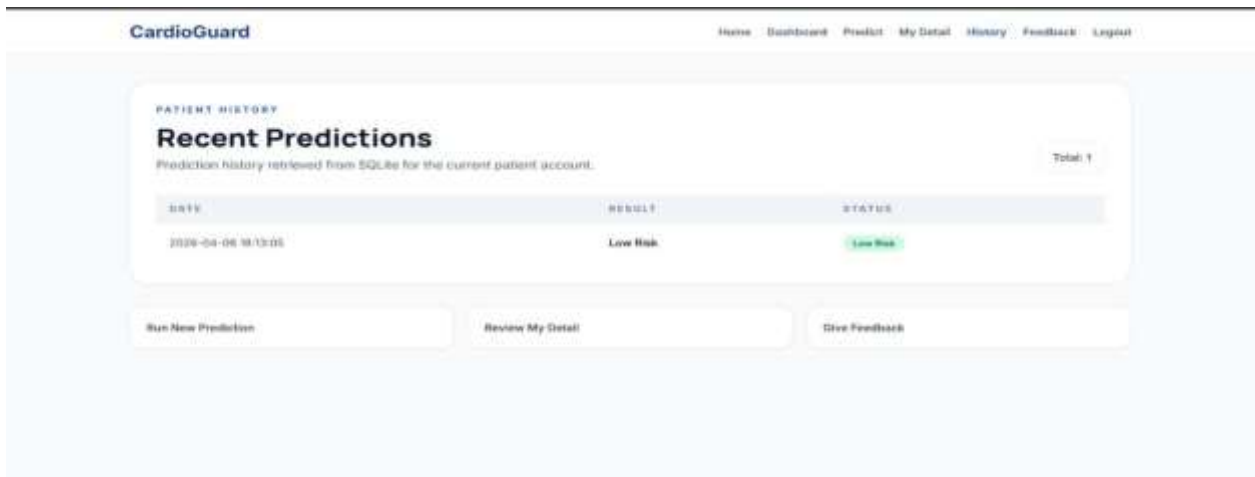


Figure 7: Implementation image 5.1.7

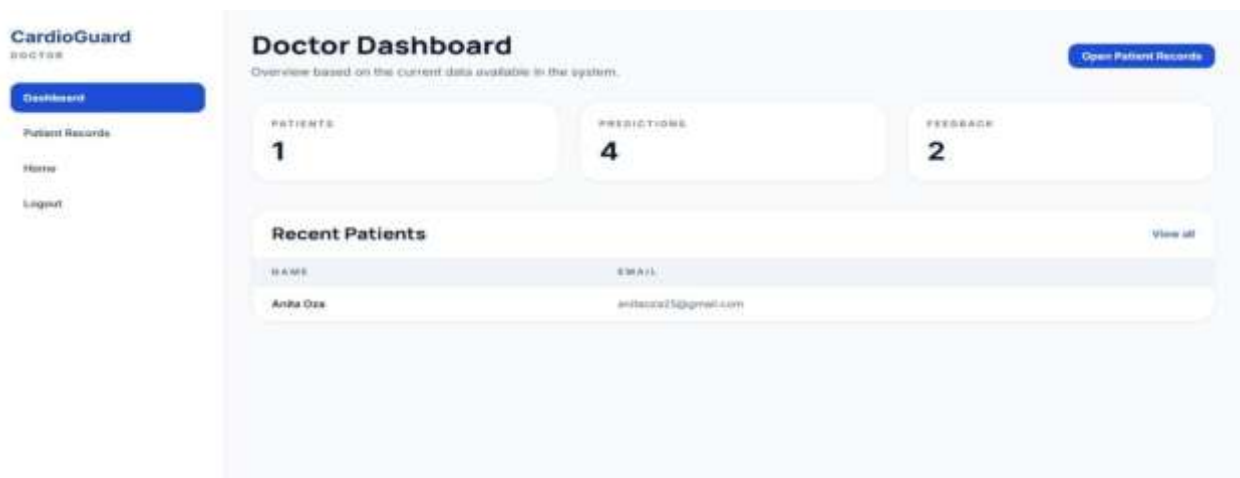


Figure 8: Implementation image 5.1.8

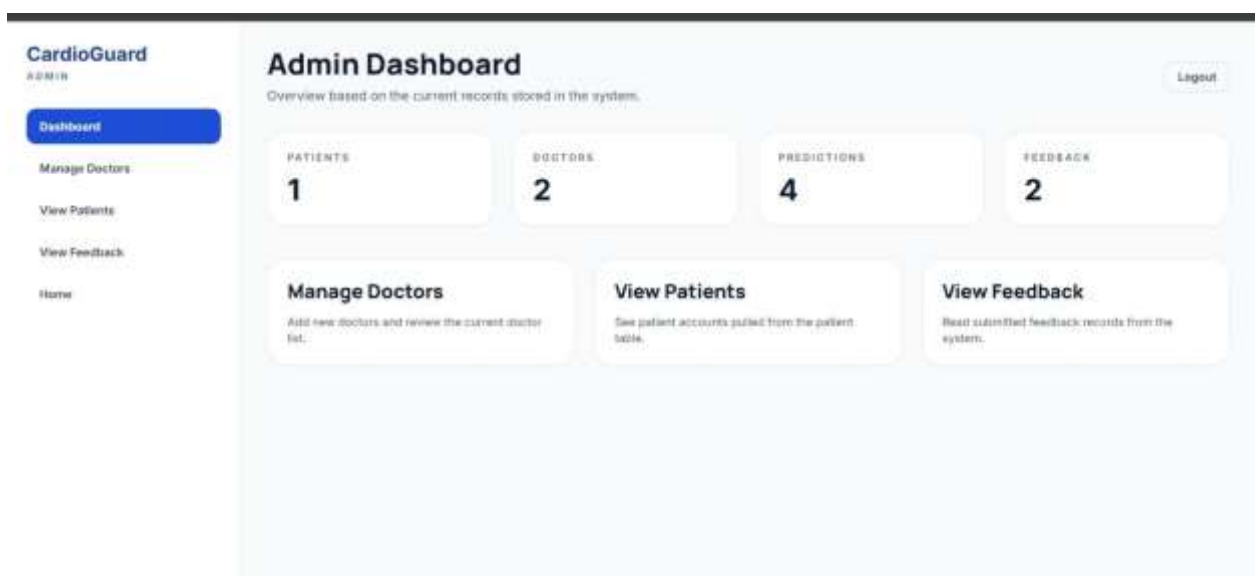


Figure 9: Implementation image 5.1.9

• Table Comparing Base Model vs Proposed System

The performance and features of the proposed *CardioGuard* system were compared with traditional heart disease prediction approaches (base models) that focus only on standalone machine learning predictions without system integration.

Table 1 : comparing Base Model and Proposed System

Criteria / Feature	Base Model (Existing Approaches)	Proposed System (CardioGuard)
Prediction Capability	Only prediction output	Prediction + complete system
Accuracy	80% – 88%	~91% (Random Forest)
User Interface	Not available	Web-based UI (Frontend)
Real-Time Prediction	Limited	Yes (instant results)
Role-Based Access	Not supported	Patient / Doctor / Admin
Data Storage	Not available	SQLite database storage
Prediction History	Not available	Available for patients
Report Generation	Not available	PDF reports (FPDF)
Email Integration	Not available	SMTP-based email option
System Integration	Not integrated	Full-stack application
Scalability	Limited	Extendable and modular

• Differences in Accuracy, Complexity, and Output

- Accuracy:**
Traditional models used in earlier research achieved moderate accuracy ranging between 80% to 88%. In contrast, the proposed system achieved approximately **91% accuracy** using the Random Forest algorithm, due to better preprocessing and ensemble learning.
- Complexity:**
Base models are simple and focus only on prediction logic. However, the proposed system is more comprehensive as it includes frontend, backend, database, and model integration. While this increases system complexity, it significantly enhances usability and real-world applicability.
- Output:**
Existing systems typically provide only a basic prediction result (Yes/No or Risk/No Risk). In comparison, the cardio Guard system provides detailed output including:
 - Risk classification (High Risk / Low Risk)
 - Downloadable PDF report
 - Stored prediction history
 - Optional email delivery of reports
- Usability and Practical Application:** The proposed system goes beyond prediction by offering a complete healthcare support platform with dashboards for patients, doctors, and administrators, making it more practical for real-world use.

VII. ADVANTAGES

- **High Prediction Accuracy:** The system uses the Random Forest algorithm, achieving approximately 91% accuracy, which provides reliable and consistent predictions.
- **User-Friendly Interface:** Developed using HTML and Tailwind CSS, the system offers a simple and intuitive interface suitable for both technical and non-technical users.
- **Real-Time Prediction:** The application provides instant results based on user input, enabling quick preliminary health assessment.
- **Role-Based Access Control:** The system supports patient, doctor, and admin roles, ensuring secure and organized access to different functionalities.
- **Prediction History Management:** Users can view and track their previous prediction results, helping in monitoring health trends over time.
- **PDF Report Generation:** The system generates downloadable reports, allowing users to maintain and share medical records easily.
- **Secure Data Handling:** User data is protected using crypt-based password hashing, ensuring privacy and security.
- **Integrated System Functionality:** Unlike basic models, the system combines machine learning with a complete web application including database storage and dashboards.
- **Additional Features:** Features like email report delivery, doctor dashboards, and admin analytics enhance usability and real-world application.

VIII. LIMITATIONS

- **Dependence on Dataset Quality:** The accuracy of the system depends on the quality, size, and diversity of the training dataset. Limited or biased data can affect prediction reliability.
- **Limited Feature Set:** The model uses only structured clinical parameters and does not include advanced medical data such as imaging (ECG graphs, scans) or genetic information.
- **Not a Medical Diagnosis Tool:** The system provides only risk prediction and cannot replace professional medical diagnosis or expert consultation.
- **Model Generalization Issues:** The trained model may not perform equally well for all populations, as it is based on specific datasets that may not cover all demographics.
- **Scalability Constraints:** The use of SQLite database is suitable for small-scale applications but may not handle large-scale or enterprise-level data efficiently.
- **Lack of Real-Time Data Integration:** The system does not currently support real-time health data from wearable devices or IoT sensors.
- **Limited Explainability:** The system provides prediction results but does not clearly explain the reasoning behind predictions, which is important in healthcare applications.

IX. FUTURE WORK

The proposed *Cardio Guard* system can be enhanced by integrating advanced machine learning techniques such as deep learning models (ANN) and improving feature selection by including lifestyle, genetic, and detailed medical data. Model optimization methods like hyperparameter tuning and cross-validation can further improve accuracy. Additionally, incorporating explainable AI can help users understand predictions, while advanced security features can strengthen data protection.

The system can also be extended by developing a mobile application for better accessibility and integrating IoT devices for real-time health monitoring. Cloud deployment can improve scalability, and the system can be expanded to support prediction of multiple diseases. Future improvements may also include advanced analytics dashboards and integration with hospital systems for practical clinical use.

X. CONCLUSION

The *Cardio Guard* system successfully implements machine learning for heart disease risk prediction through a full-stack web application. Multiple algorithms were trained and evaluated, with the Random Forest model achieving the highest accuracy and being selected for deployment. The system integrates a Flask backend, responsive frontend, and SQLite database to provide real-time prediction, classification into high-risk and low-risk categories, and downloadable PDF reports, along with prediction history and role-based access for patients, doctors, and administrators. Practically, the system serves as an effective preliminary screening tool for early detection of heart disease and assists users and healthcare professionals in making informed decisions, demonstrating the real-world applicability of machine learning in healthcare.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Prof. S. N. Mulhar, Assistant Professor, Department of Computer Engineering, STES'S Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, for her continuous guidance, valuable suggestions, and support throughout the successful completion of this project. Her mentorship played a crucial role in shaping the development and implementation of the CardioGuard system.

The authors also extend their heartfelt thanks to the Department of Computer Engineering, STES'S Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, for providing the necessary resources, infrastructure, and academic environment required to complete this project successfully. Their support and encouragement made it possible to transform the research work into a fully functional system.

REFERENCES

- [1] J. Smith, A. Brown, "Machine Learning-Based Heart Disease Prediction Using Ensemble Methods," IEEE Access, vol. 13, pp. 10234–10245, 2025.
- [2] R. Verma, S. Kulkarni, "Explainable AI for Cardiovascular Risk Prediction Using Hybrid Models," Springer Healthcare Informatics, 2025.
- [3] L. Chen, Y. Zhao, "Advanced Random Forest Optimization for Medical Diagnosis Systems," Elsevier Journal of Biomedical Informatics, 2025.
- [4] P. Sharma, K. Singh, "Web-Based Healthcare Systems Using Machine Learning and Flask Framework," International Journal of Computer Applications, 2025.
- [5] M. Gupta, R. Shah, "Comparative Analysis of Machine Learning Algorithms for Heart Disease Detection," IEEE Xplore, 2025.
- [6] A. Khan, M. Ali, "Heart Disease Prediction Using Supervised Learning Techniques," Journal of Healthcare Engineering, 2024.
- [7] S. Patel, D. Mehta, "Integration of Machine Learning Models in Web Applications for Disease Prediction," Springer, 2024.
- [8] Y. Li, X. Wang, "Performance Evaluation of SVM and Random Forest in Medical Data Classification," Elsevier, 2024.
- [9] N. Gupta, P. Agarwal, "Flask-Based Web Systems for Real-Time Healthcare Prediction," International Journal of Advanced Research in Computer Science, 2024.
- [10] T. Roy, A. Das, "Data Preprocessing Techniques for Improving ML Model Accuracy in Healthcare," IEEE Access, 2024.

- [11] K. Ramesh, V. Iyer, "Machine Learning Models for Early Heart Disease Detection," Journal of Medical Systems, 2023.
- [12] S. Ahmed, R. Khan, "Ensemble Learning Techniques for Disease Prediction," Springer Nature, 2023.
- [13] M. Singh, A. Kaur, "Prediction of Cardiovascular Disease Using KNN and Logistic Regression," IEEE Conference Proceedings, 2023.
- [14] P. Desai, N. Shah, "Role-Based Web Applications in Healthcare Using Flask," International Journal of Engineering Research, 2023.
- [15] H. Lee, J. Kim, "Evaluation Metrics for Machine Learning Models in Healthcare Applications," Elsevier, 2023.
- [16] A. Verma, S. Jain, "Heart Disease Prediction Using Decision Tree and Random Forest," IEEE Xplore, 2022.
- [17] R. Patel, M. Joshi, "Implementation of Machine Learning in Healthcare Systems," Springer, 2022.
- [18] S. Nair, K. Menon, "Comparative Study of ML Algorithms for Disease Prediction," Journal of Computer Science, 2022.
- [19] D. Kumar, P. Mishra, "Use of Data Mining Techniques in Medical Diagnosis," IEEE, 2022.
- [20] A. Choudhary, R. Singh, "Web-Based Medical Systems with Integrated ML Models," IJERT, 2022.
- [21] UCI Machine Learning Repository, "Heart Disease Dataset," 2021.
- [22] Kaggle, "Cardiovascular Disease Dataset," 2021.
- [23] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 2021.
- [24] M. Grinberg, "Flask Web Development: Developing Web Applications with Python," O'Reilly Media, 2021.
- [25] R. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," 2020.