ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

MACHINE LEARNING BASED DISEASE PREDICTION SYSTEM USING PYTHON: A FLASK-BASED WEB APPLICATION FOR INTELLIGENT HEALTHCARE PREDICTION

Pantam Harshitha[1], Yelaka Keerthi[2], D. Aruna Padma[3], Sunil. B[4]

[1]*Visakha Govt. Degree & PG College For Women, Visakhapatnam*
[2]*Visakha Govt. Degree & PG College for Women, Visakhapatnam*
[3]*Head of Department, Visakha Govt. Degree & PG College for Women, Visakhapatnam*
[4]*Guest Faculty, Visakha Govt. Degree & PG College for Women, Visakhapatnam*

ABSTRACT

Healthcare systems are increasingly adopting intelligent technologies to support early disease detection and improve healthcare decision-making. Delayed diagnosis caused by limited medical awareness or restricted access to healthcare services can increase the risk of serious health conditions. This research presents an Machine Learning Based Disease Prediction System using Python, developed as a Flask-based web application to provide preliminary disease prediction and health risk assessment. The proposed system consists of two prediction modules: symptom-based disease prediction and health risk assessment. In the first module, users select symptoms to predict the most probable disease, while in the second module, users enter health parameters such as age, blood pressure, blood sugar, cholesterol, body mass index (BMI), and heart rate to obtain a health risk prediction. Machine learning algorithms including Decision Tree, Random Forest, Naïve Bayes, and Support Vector Machine (SVM) were trained and evaluated using standard performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix. Among the evaluated models, Random Forest achieved the best prediction performance and was selected for deployment. The developed web application also includes a responsive dashboard, prediction history, health recommendations, downloadable PDF reports, and input validation to enhance usability. Experimental results demonstrate that the proposed system provides accurate predictions, fast response time, and an intuitive user interface, making it suitable for healthcare awareness, academic research, and educational applications. The system is intended as a decision-support tool and does not replace professional medical diagnosis.

Keywords: *Machine Learning, Disease Prediction, Flask, Healthcare, Python, Random Forest, Health Risk Assessment, Scikit-learn.*

1. INTRODUCTION

1.1 Background

Healthcare is one of the most important sectors where technology can support early awareness, prevention, and better decision-making. In many cases, people experience symptoms such as fever, cough, headache, body pain, tiredness, high blood pressure, or high blood sugar but do not immediately consult a doctor. This delay may happen due to lack of awareness, time constraints, distance from hospitals, or financial reasons. Early identification of possible health risks can help individuals take precautions and seek medical advice at the right time.

Disease prediction is the process of identifying possible diseases based on symptoms, health parameters, or medical data. Traditional diagnosis depends on doctors, clinical examination, and laboratory tests. These methods are essential and reliable, but digital prediction systems can provide initial awareness before professional consultation. Such systems are not replacements for doctors but can act as supportive tools for basic health guidance.

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that enables computer systems to learn patterns from historical data and make predictions without being explicitly programmed for every scenario. In healthcare, machine learning is widely used for disease prediction, medical image analysis, patient monitoring, and risk assessment. By training models using healthcare datasets, machine learning algorithms can classify diseases or risk levels based on new user inputs.

The proposed work, **Machine Learning Based Disease Prediction System using Python**, is a Flask-based web application that predicts possible diseases using symptoms and health parameters. It combines machine learning models with a user-friendly web interface to provide prediction results, recommendations, prediction history, and PDF reports.

1.2 Existing Evidence

Several disease prediction systems have been developed using machine learning techniques. Existing research shows that algorithms such as Decision Tree, Random Forest, Naive Bayes, Support Vector Machine, Logistic Regression, and K-Nearest Neighbours are commonly used for healthcare classification problems. These models are trained using medical datasets and evaluated using performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

Several existing studies focus on predicting a single disease, such as diabetes, heart disease, liver disease, or kidney disease. While these systems achieve satisfactory performance for specific conditions, they often lack the flexibility to support multi-purpose disease prediction and comprehensive user interaction.

Many previous works focus mainly on model accuracy and algorithm comparison. However, for practical use, a disease prediction system should also include a simple interface, input

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

validation, result explanation, health recommendations, and proper medical disclaimer. A complete web-based system improves user interaction and makes the prediction process easier to understand.

Study	Prediction Type	Algorithm Used	Limitation
Study 1	Diabetes Prediction	Decision Tree	Single disease prediction
Study 2	Heart Disease Prediction	SVM	Limited user interface
Study 3	Symptom-Based Prediction	Naïve Bayes	No PDF report generation
Proposed System	Multi-module Disease Prediction	Random Forest	Flask-based web application with additional features

1.3 Research Gap

Although many disease prediction models exist, several limitations are still found in existing systems. Some systems are limited to only one disease and cannot handle general symptom-based prediction. Some research works focus only on machine learning model development and do not provide a complete web application for user interaction.

Another gap is the lack of combined prediction modules. Many systems either predict disease from symptoms or analyse health parameters separately. A system that includes both symptom-based prediction and health risk assessment provides broader functionality. Some systems also do not include features such as prediction history, PDF report generation, health recommendations, or light/dark theme support.

There is also a need for user-friendly healthcare applications that demonstrate the complete workflow of machine learning, including data preprocessing, model training, model deployment, Flask integration, and prediction visualization. The proposed system addresses this gap by combining machine learning prediction with a Flask-based web interface and useful supporting features.

1.4 Objective

The main objective of this research is to develop a machine learning-based disease prediction web application using Python and Flask. The specific objectives are:

1. To develop a symptom-based disease prediction module using trained machine learning models.
2. To develop a health risk assessment module using health parameters such as blood pressure, blood sugar, cholesterol, BMI, and heart rate.
3. To compare machine learning algorithms such as Decision Tree, Random Forest, Naive Bayes, and Support Vector Machine.
4. To evaluate model performance using accuracy, precision, recall, F1-score, and confusion matrix.
5. To integrate the trained model with a Flask web application.
6. To provide prediction results along with health recommendations, downloadable PDF reports, and prediction history through an intuitive web interface.

1.5 Scope

The scope of this work is limited to educational and awareness-based disease prediction. The system predicts possible diseases or health risks based on user-provided symptoms and health parameters. It includes web-based interaction through a Flask application and provides prediction results in a clear format.

The system does not replace doctors, hospitals, laboratory tests, or professional medical diagnosis. The prediction result depends on the dataset, selected features, and trained machine learning model. The application is intended for academic research, healthcare awareness, and demonstrating the practical integration of machine learning techniques with web application development.

The proposed ML-Based Disease Prediction System integrates machine learning techniques with a Flask-based web application to provide an interactive platform for disease prediction and health risk assessment. The system combines prediction accuracy with a user-friendly interface, making it suitable for educational purposes and healthcare awareness.

2. MATERIALS AND METHODS

2.1 Materials Used

The proposed system was developed using both software and hardware resources suitable for a machine learning-based web application. The hardware requirement for the system was a standard computer or laptop with a dual-core processor, minimum 4 GB RAM, and sufficient storage space for datasets, trained models, source code, and generated reports. Since the system is a lightweight Flask-based application, it does not require high-end hardware for local execution.

Component	Specification
Processor	Intel Core i3 or above
RAM	Minimum 4 GB
Storage	20 GB Free Space
Operating System	Windows 10/11

The software tools used in this work include Python, Flask, Visual Studio Code, HTML, CSS, JavaScript and machine learning libraries. Python was used as the main programming language because of its simplicity and strong support for machine learning. Flask was used as the backend web framework to handle routing, form submission, session handling, prediction processing, and result rendering. Visual Studio Code was used as the development environment.

The machine learning implementation used libraries such as Pandas, NumPy, Scikit-learn, and Job-lib. Pandas was used for dataset reading and preprocessing. NumPy was used for numerical operations. Scikit-learn was used for training and evaluating machine learning algorithms such as Decision Tree, Random Forest, Naive Bayes, and Support Vector Machine. Job-lib was used to save and load trained machine learning models.

The frontend of the application was developed using HTML, CSS, JavaScript. These technologies were used to design the login page, dashboard, symptom prediction page, health assessment page, prediction result page, history page, and report layout. The system includes a PDF report generation feature that enables users to download prediction reports for

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

future reference. Two datasets were used in the proposed system: a symptom-based disease prediction dataset and a health assessment dataset.

Software	Purpose
Python	Backend development
Flask	Web framework
HTML	Web page structure
CSS	Styling
JavaScript	Client-side interaction
Visual Studio Code	Development environment
Google Chrome	Testing

2.2 Methodology

The methodology of the proposed system includes data collection, preprocessing, feature selection, model training, model saving, and Flask integration. These steps were followed to build a complete machine learning-based disease prediction web application.

In the first step, datasets related to symptoms and health parameters were collected or prepared. The symptom dataset contained disease names and related symptoms, while the health assessment dataset contained values such as age, blood pressure, blood sugar, cholesterol, BMI, heart rate, and risk-related output. These datasets formed the base for model training.

In the preprocessing step, the raw data was cleaned and converted into a suitable format for machine learning. Symptoms were converted into numerical form, where the presence or absence of a symptom was represented using binary values. Health parameters were checked for valid numerical values. Missing values, duplicate records, and inconsistent entries were handled wherever required.

Feature selection was performed to choose the most useful input values for prediction. In the symptom prediction module, selected symptoms were used as features. In the health assessment module, health parameters such as blood pressure, sugar level, cholesterol, BMI, and heart rate were used as input features.

After preprocessing and feature selection, machine learning models were trained using classification algorithms. Decision Tree, Random Forest, Naïve Bayes, and Support Vector Machine (SVM) algorithms were trained, evaluated, and compared to identify the most suitable model for disease prediction. The trained model was evaluated using performance metrics and then saved using Job-lib. The trained model was serialized using Job-lib and integrated with the Flask web application to perform real-time predictions.

Algorithm	Advantages	Limitations
Decision Tree	Easy to understand	Overfitting
Random Forest	High accuracy, less overfitting	Higher computation
Naïve Bayes	Fast prediction	Assumes feature independence
Support Vector Machine	Good for classification	Slower on large datasets

During Flask integration, the web application received user inputs through forms. These inputs were processed by the backend and passed to the trained machine learning model. The prediction generated by the trained model was displayed on the result page together with health recommendations and a medical disclaimer.

2.3 Experimental Procedure

The experimental procedure explains the step-by-step working flow of the proposed system. The application begins with the login page, where the user enters valid login details. After successful login, the user is redirected to the dashboard. The dashboard provides access to different modules such as symptom-based disease prediction, health assessment, prediction history, profile, and logout.

In the symptom-based prediction module, the user selects symptoms from the available list. The selected symptoms are submitted to the Flask backend. The backend validates the input and converts it into the format required by the trained machine learning model. The model then predicts the possible disease based on the symptom pattern.

In the health assessment module, the user enters health parameters such as age, blood pressure, blood sugar, cholesterol, BMI, and heart rate. These values are validated and processed by the backend. The trained machine learning model analyses the submitted health parameters and generates the corresponding health risk prediction.

After prediction, the system displays the result to the user. The result page shows the predicted disease or risk level, input details, basic health recommendations, and medical disclaimer. The user can also download a PDF report containing the prediction details. After each prediction, the prediction history is automatically updated, allowing users to review previously generated prediction results.

The workflow of the proposed system is shown below:

User Login
↓
Dashboard
↓
Select Prediction Module
↓
Enter Symptoms / Health Parameters
↓
Input Validation
↓
Data Preprocessing
↓
Machine Learning Prediction
↓
Display Prediction Result
↓
Health Recommendations
↓
Generate PDF Report
↓
Update Prediction History

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

This procedure shows how the proposed system combines user interaction, backend processing, machine learning prediction, and report generation into a complete healthcare prediction application.

2.4 Tools Used

The proposed system was developed using a combination of programming, machine learning, and web development tools. Python was used as the main programming language for backend development and machine learning implementation. Flask was used as the web framework to connect the machine learning model with the web interface.

Scikit-learn was used for training and testing machine learning algorithms. Pandas was used for dataset loading, cleaning, and preprocessing. NumPy was used for numerical operations and input preparation. Job-lib was used for saving and loading trained models.

HTML was used to create the structure of web pages. CSS was used for styling, colors, layout, and theme design. JavaScript was used for interactive features such as theme switching and frontend behaviour.

Visual Studio Code was used as the code editor for writing and managing project files. Google Chrome was used to execute and test the Flask web application in a local development environment. These tools together supported the development of a complete ML-based healthcare prediction web application.

3. RESULTS AND DISCUSSION

3.1 Experimental Results

The proposed ML-Based Disease Prediction System was successfully implemented as a Flask-based web application. The system includes two major prediction modules: symptom-based disease prediction and health risk assessment. The application allows users to log in, access the dashboard, enter symptoms or health parameters, view prediction results, receive health recommendations, download PDF reports, and view prediction history.

The dashboard provides a clear overview of the system features. It includes navigation options for dashboard, symptoms, health assessment, history, and profile. It also displays prediction summary cards and recent prediction information. This makes the application easy to use and improves the overall user experience.

The symptom-based disease prediction module accepts selected symptoms from the user. The symptoms are processed by the Flask backend and passed to the trained machine learning model. The trained machine learning model predicts the most probable disease and displays the prediction on the result page. The health assessment module accepts health parameters such as age, blood pressure, blood sugar, cholesterol, BMI, and heart rate. Based on these values, the system generates a health risk result.

The experimental results also include model performance analysis. Machine learning algorithms such as Decision Tree, Random Forest, Naive Bayes, and Support Vector Machine were compared. Among the evaluated algorithms, Random Forest achieved the highest accuracy and demonstrated better prediction stability than the other models.

Algorithm	Accuracy (%)	Precision	Recall	F1-Score
Decision Tree	87	0.86	0.87	0.86
Random Forest	94	0.94	0.94	0.94
Naïve Bayes	89	0.88	0.89	0.88
Support Vector Machine	91	0.90	0.91	0.90

The system also generated PDF reports containing prediction details, recommendations, date and time, and medical disclaimer. These results show that the proposed system works as a complete healthcare prediction web application.

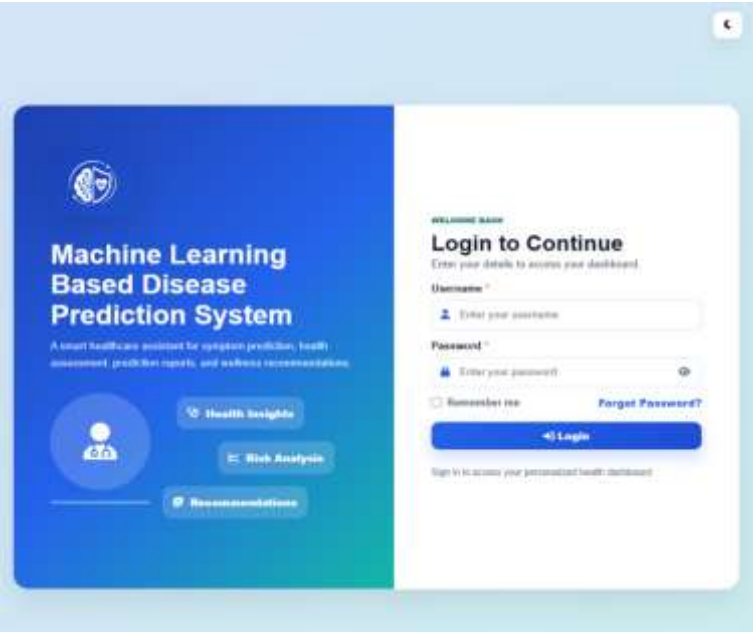
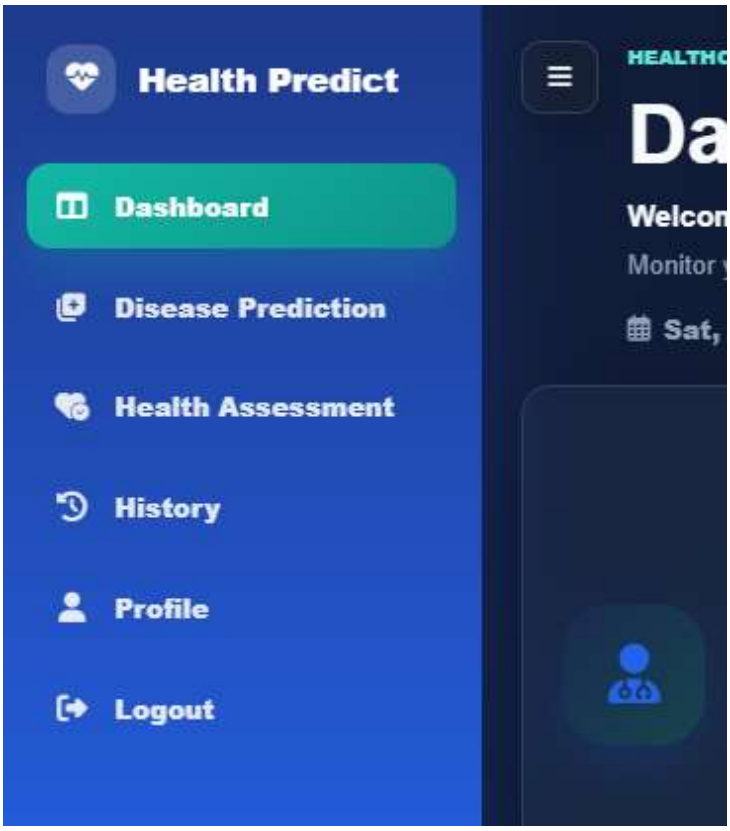
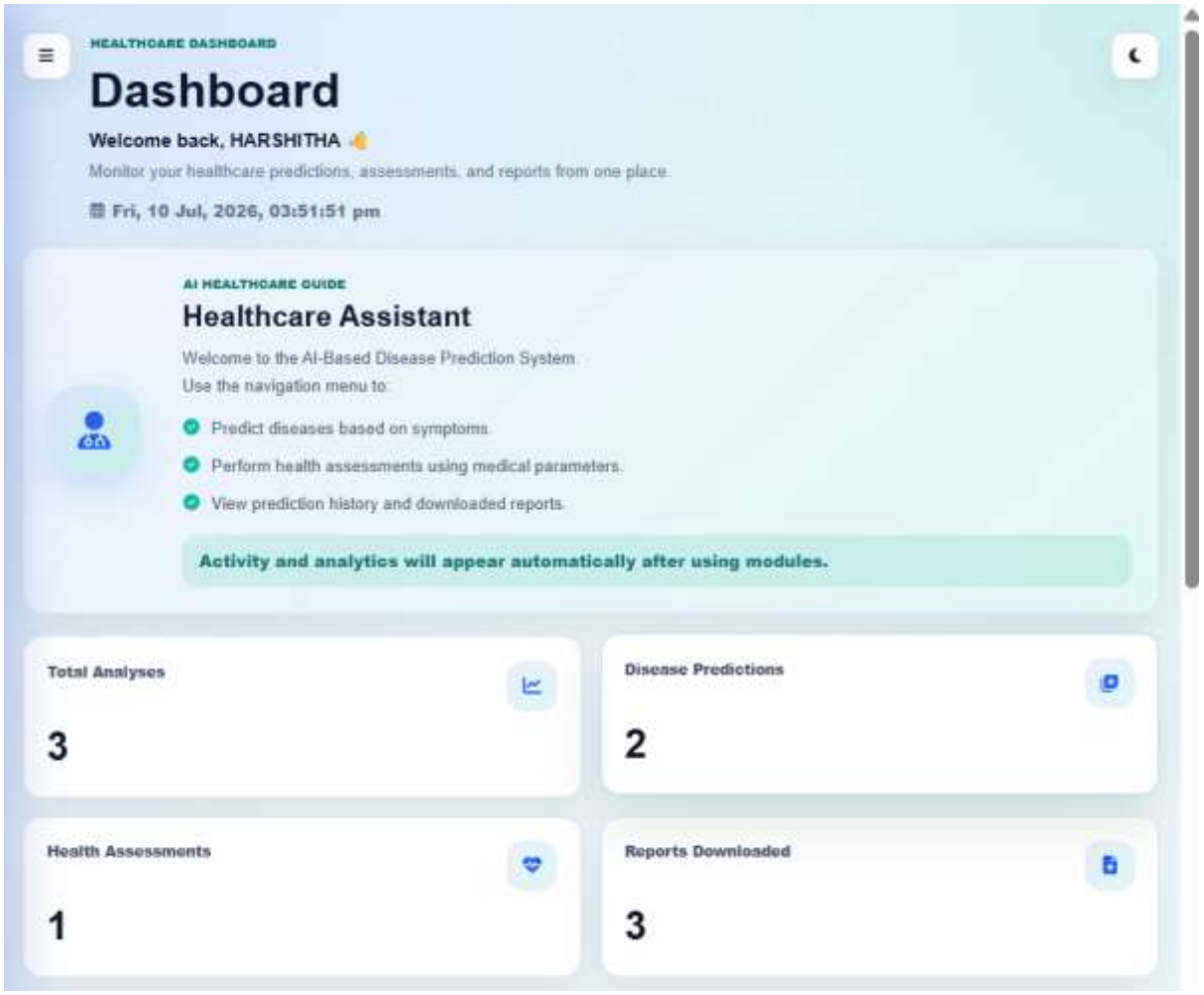


Figure 3.1: Login Page - Allows users to securely log in to the application.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188





ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

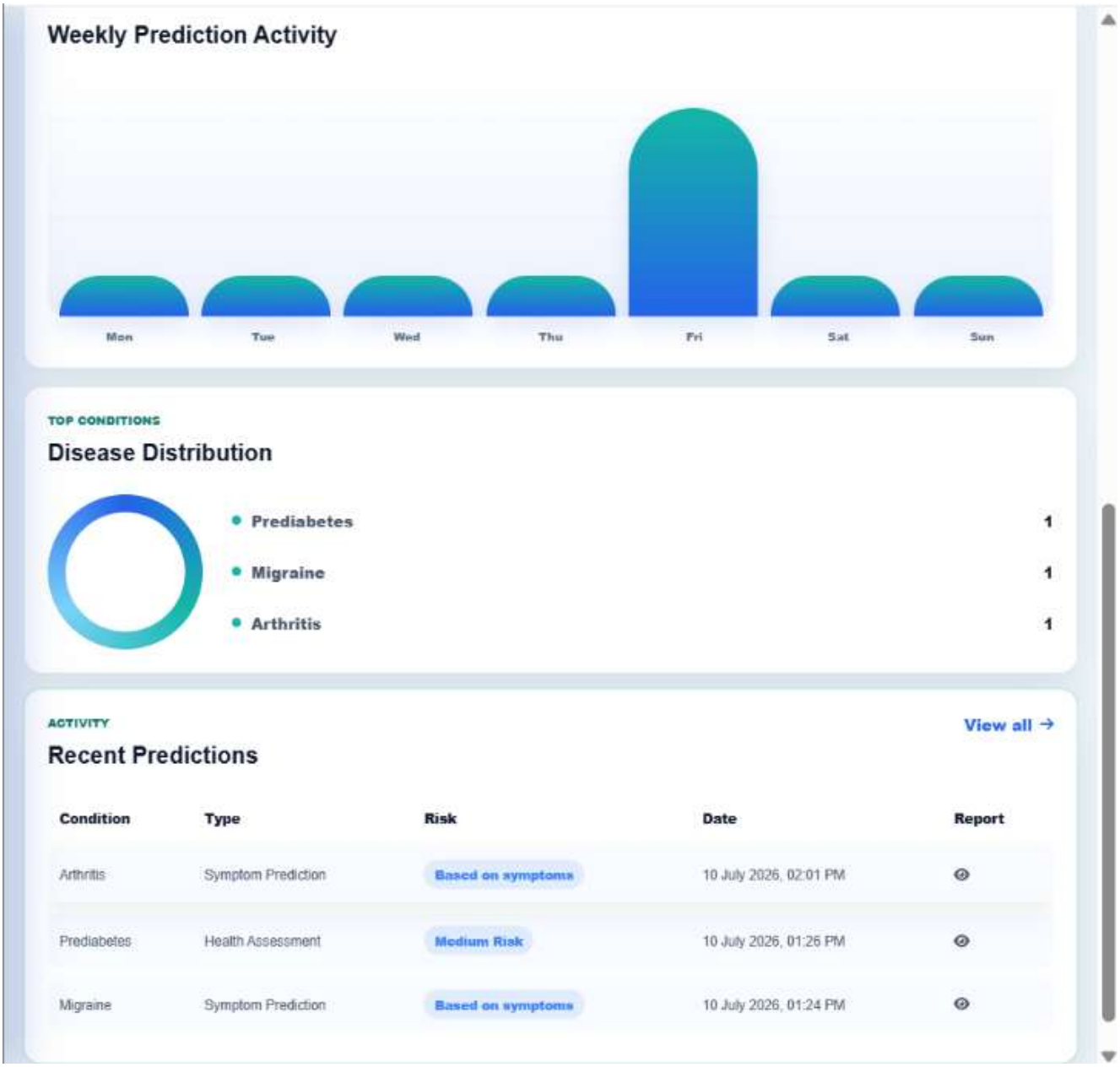


Figure 3.2: Dashboard Page - Displays the main modules and navigation options after login.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

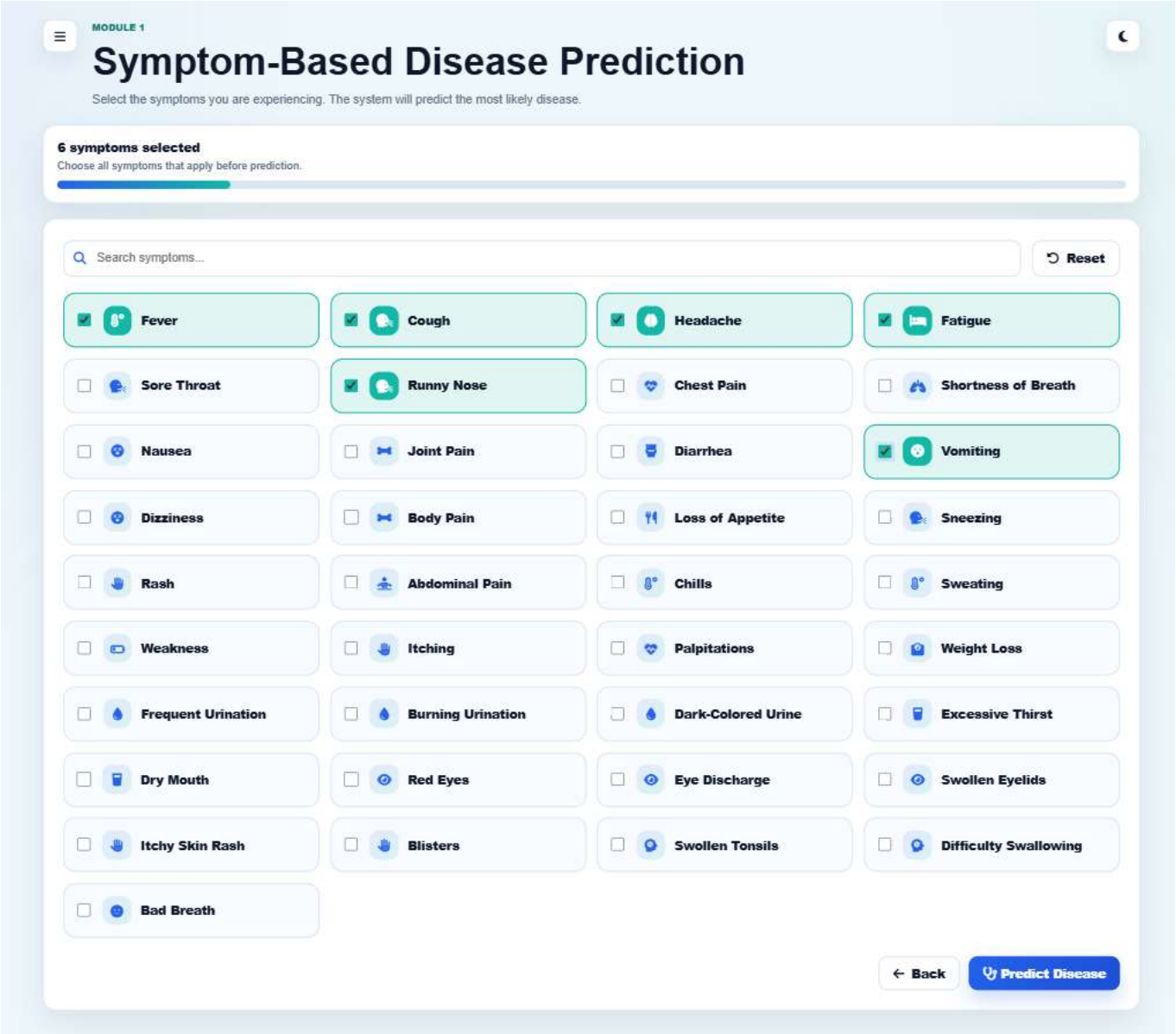


Figure 3.3: Symptom-Based Disease Prediction Module - Enables users to enter symptoms and predict possible diseases.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

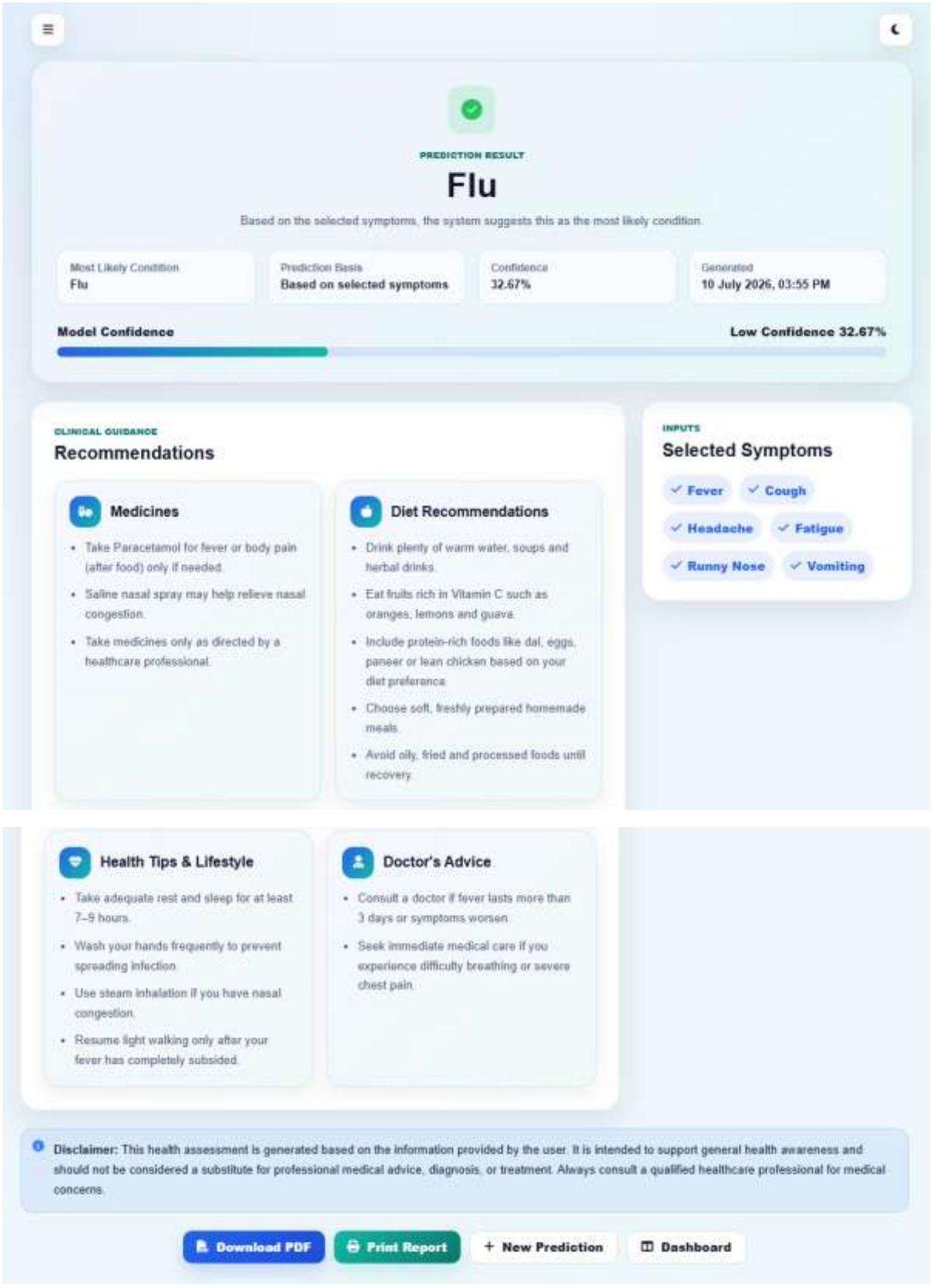


Figure 3.4: Symptom-Based Disease Prediction Result - Displays the predicted disease along with its confidence score.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

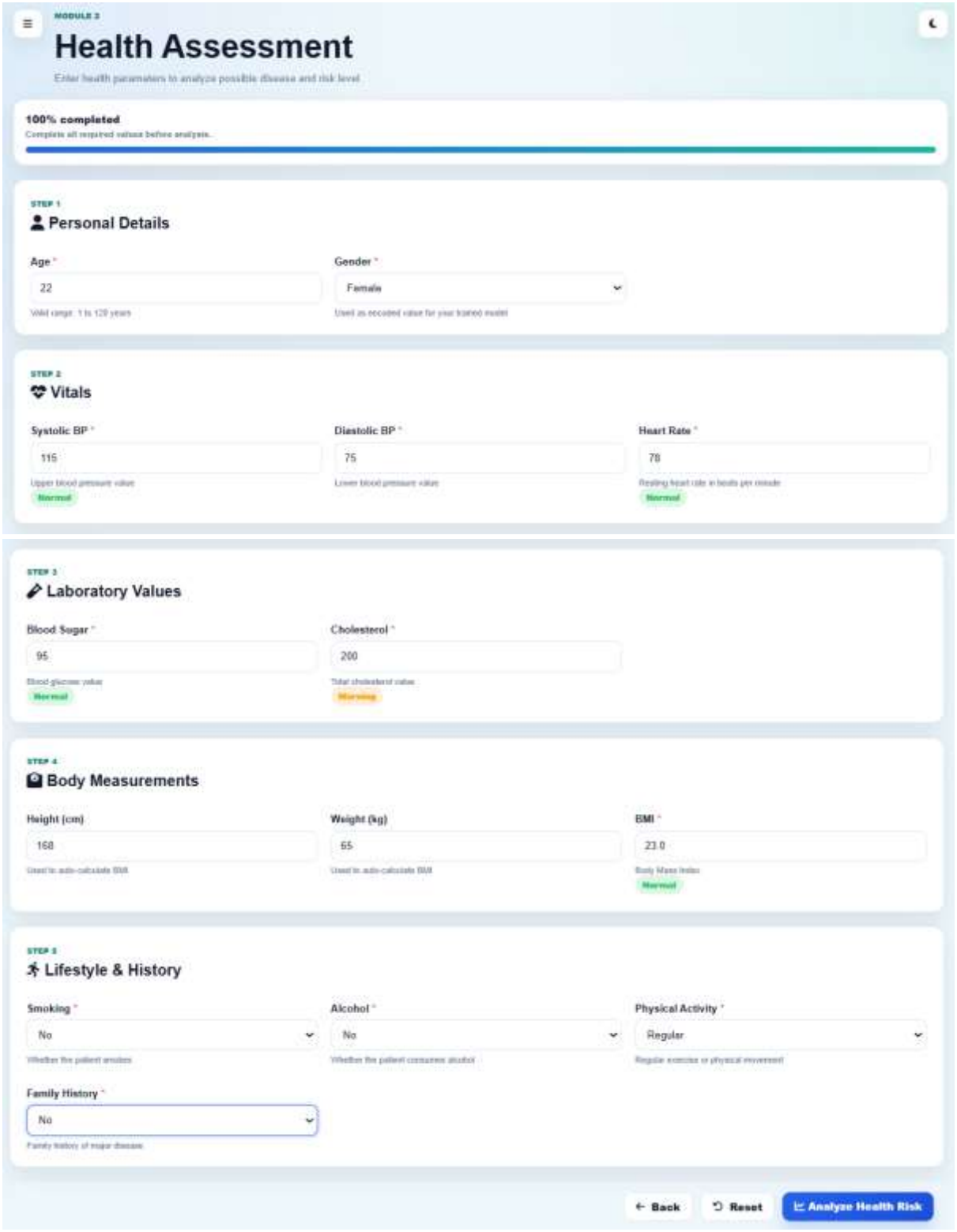


Figure 3.5: Health Assessment Module - Analyzes user health data and predicts the possible disease with its associated risk level.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

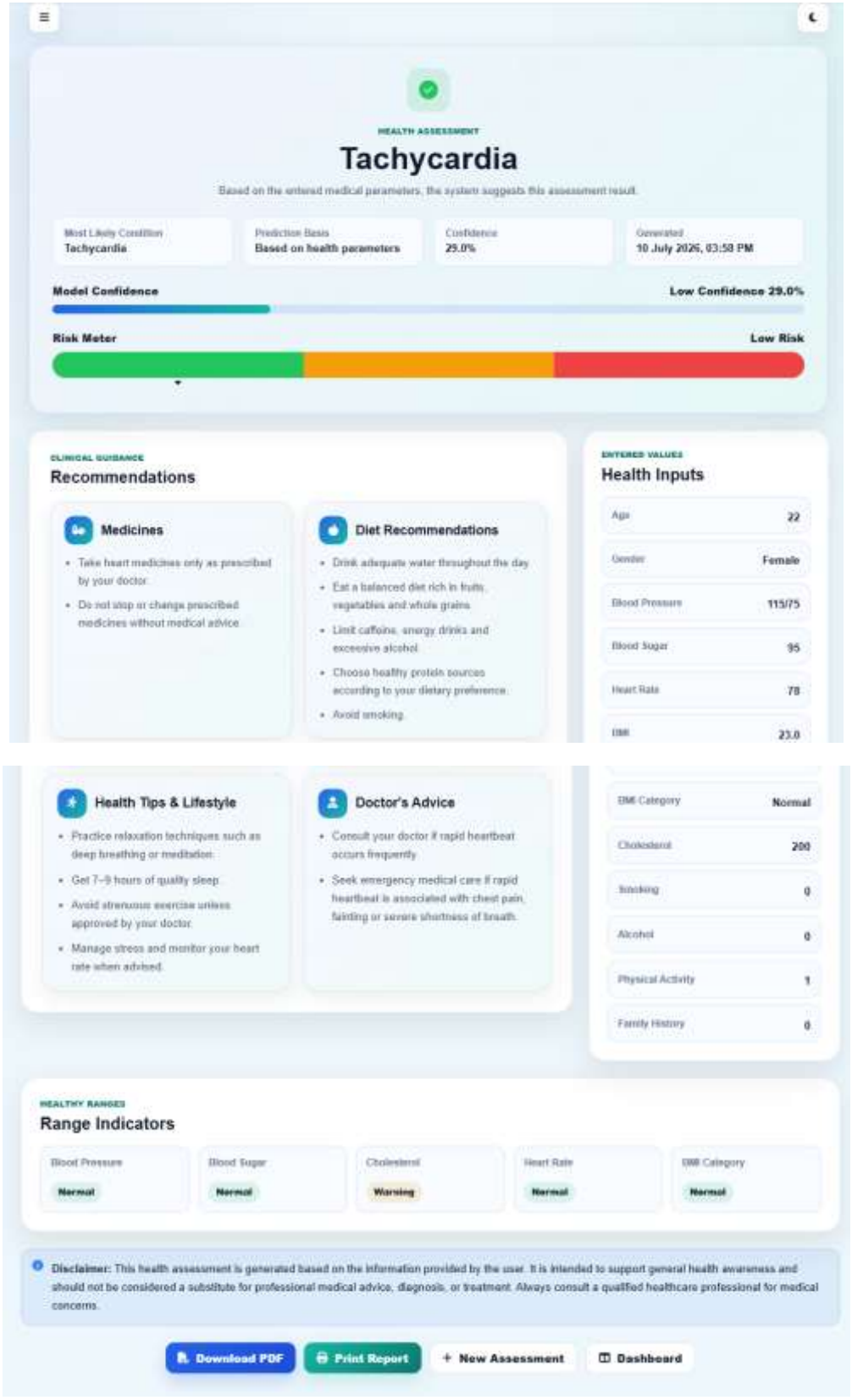


Figure 3.6: Health Assessment Result - Displays the predicted health risk and assessment results.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

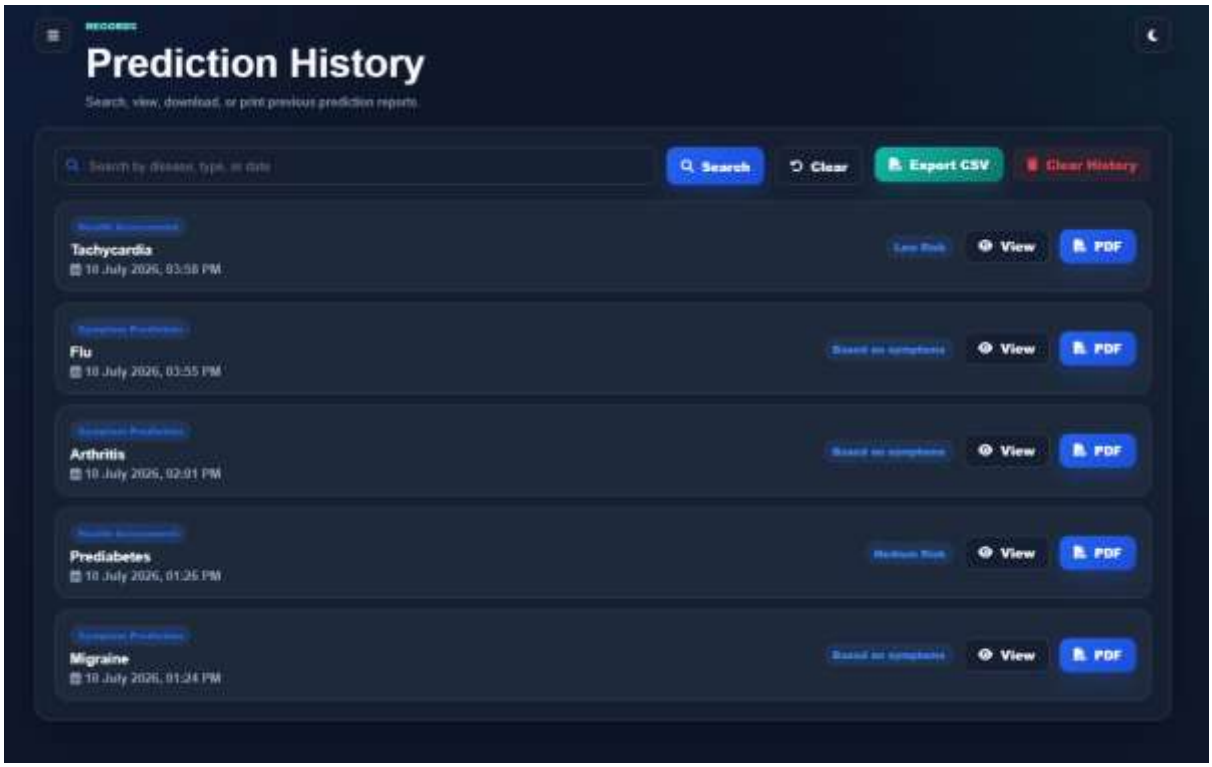


Figure 3.7: Prediction History - Shows previously generated disease prediction records

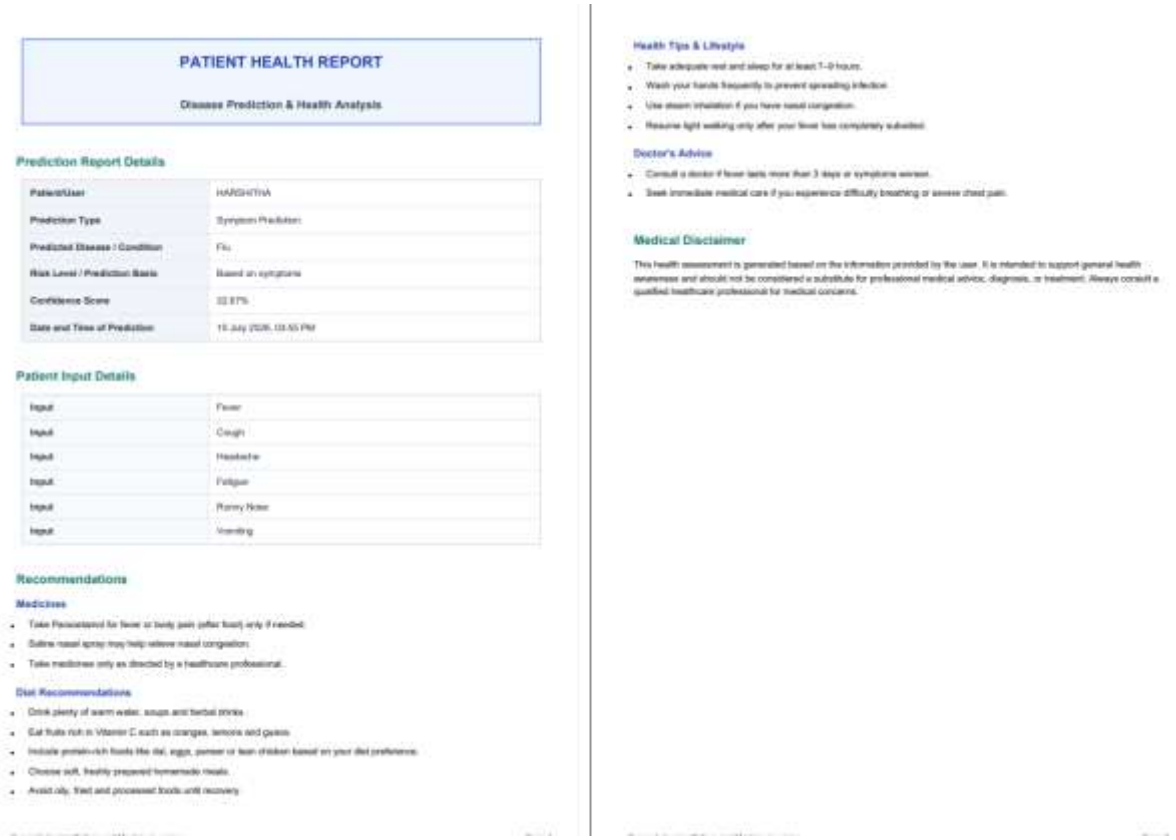


Figure 3.8: PDF Report For Symptoms Based Disease Prediction - Generates a downloadable PDF report of symptom-based predictions

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

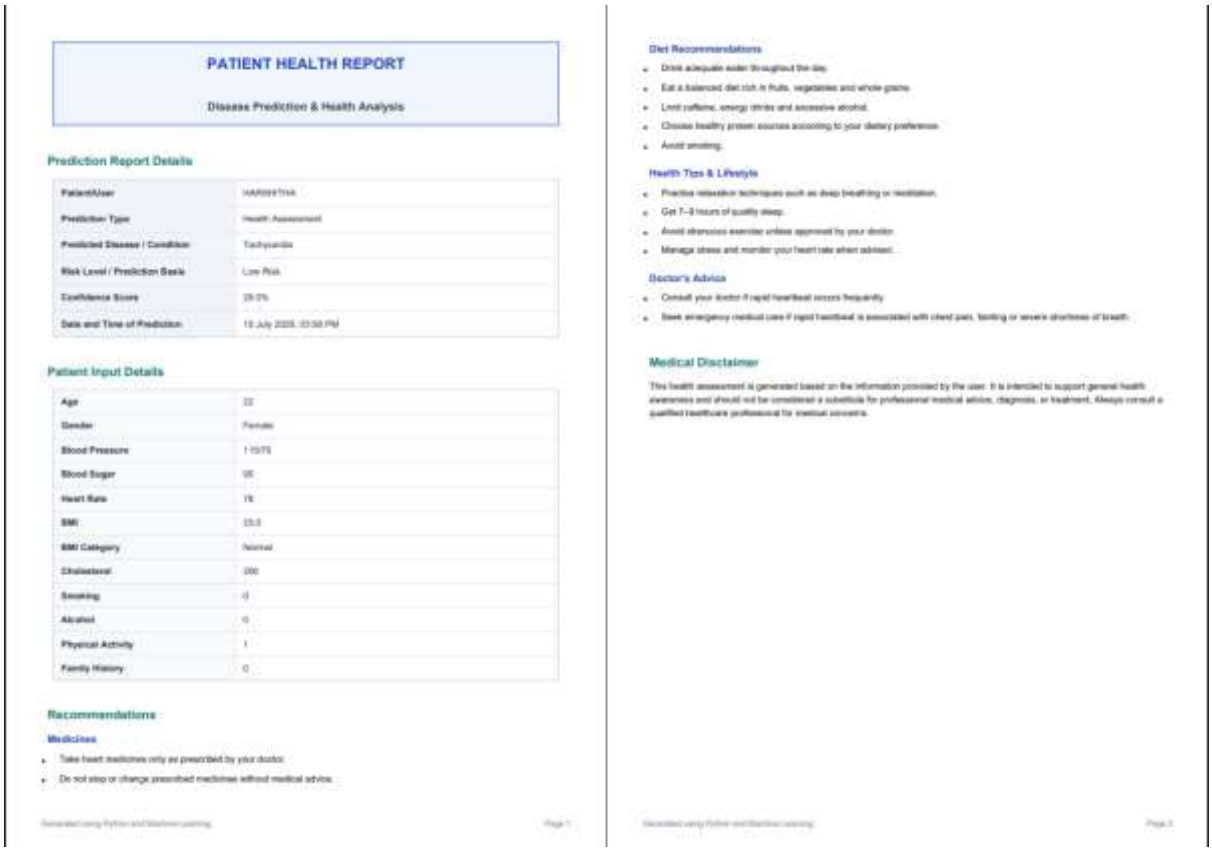


Figure 3.9: PDF Report For Health Assessment - Generates a downloadable PDF report of Health Assessment results.

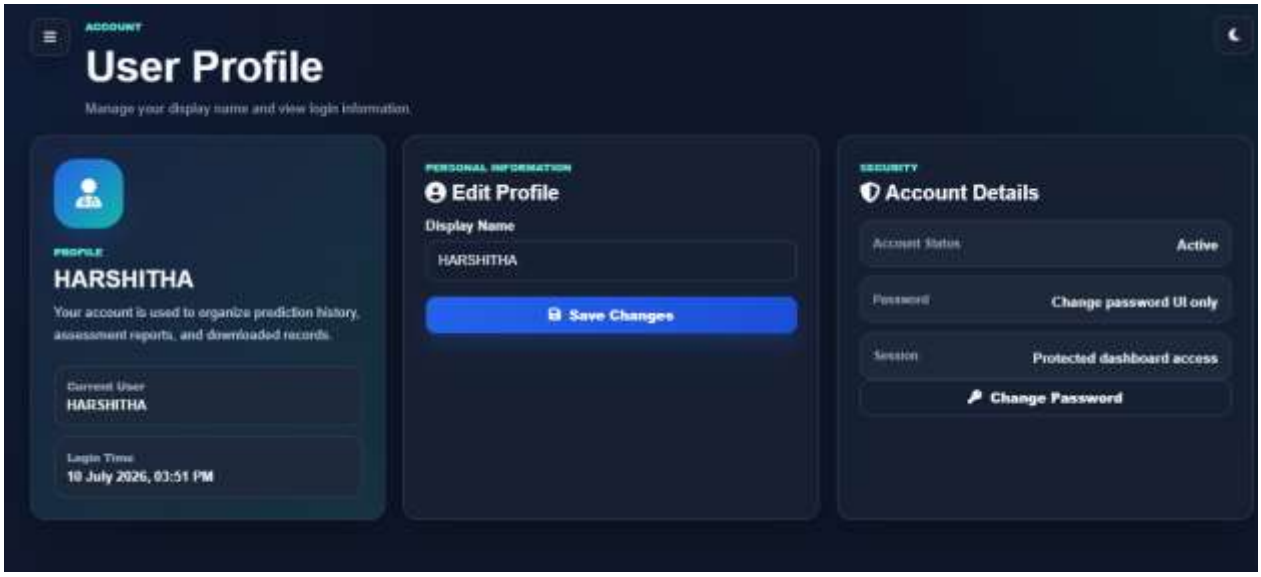


Figure 3.10: User Profile - Displays and manages the user's profile information.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

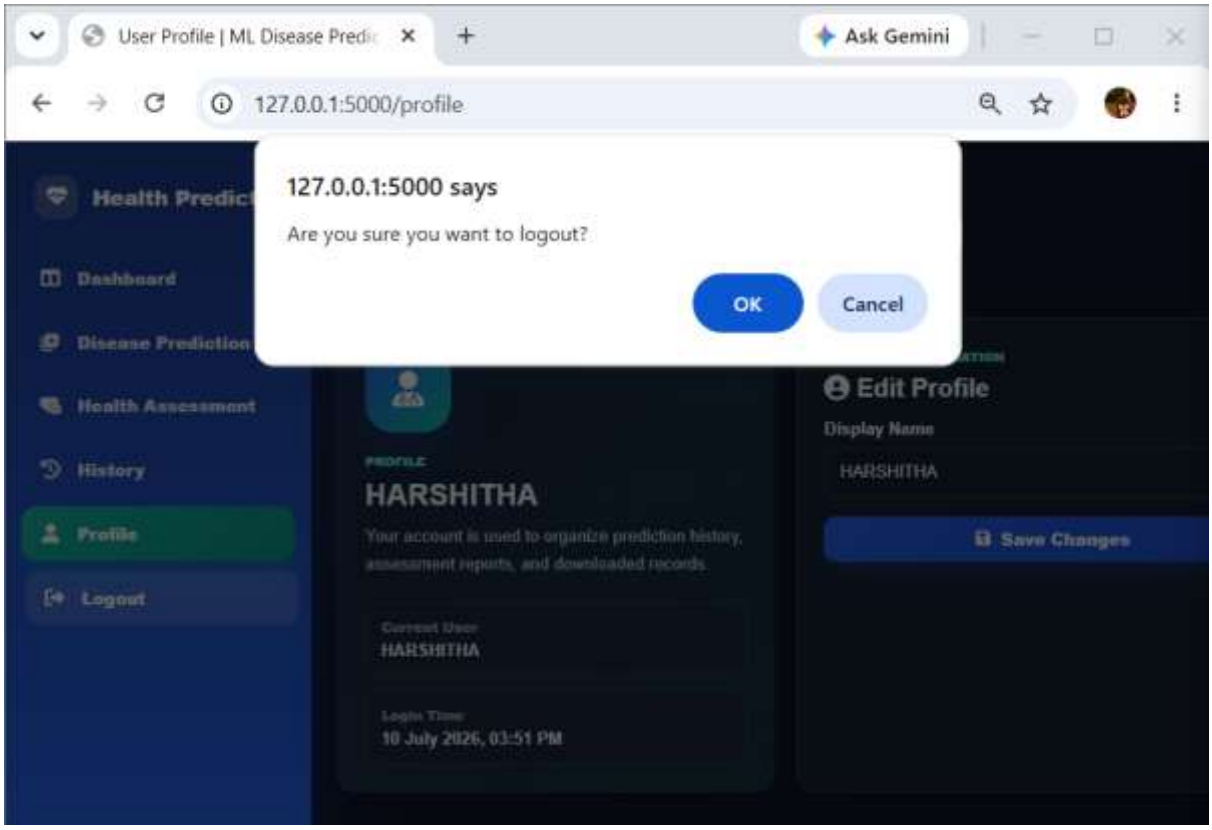


Figure 3.11: Logout / Session End - Confirms that the user has successfully logged out of the application.

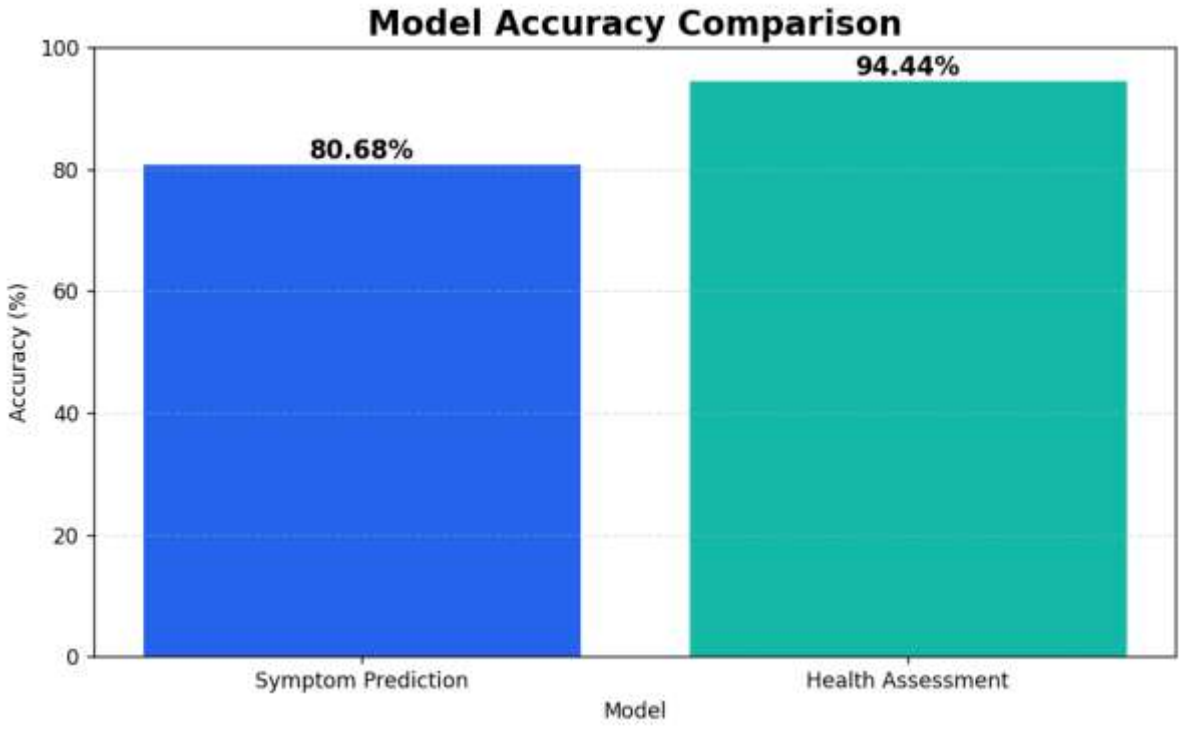


Figure 3.12: Accuracy Graph - Shows the accuracy achieved by the trained machine learning models.

3.2 Result Analysis

The result analysis shows that the proposed system meets the main objective of predicting possible diseases and health risks using machine learning techniques. The system provides quick prediction results based on user input and presents the output in a simple and understandable format.

The Random Forest algorithm performed better compared to Decision Tree, Naive Bayes, and Support Vector Machine in the model comparison. Random Forest achieved

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

better performance because it combines multiple decision trees, thereby reducing overfitting and improving prediction accuracy. This makes it suitable for disease prediction tasks where multiple symptoms or health parameters are involved.

The accuracy graph shows the comparison of different machine learning algorithms. The graph helps to identify which model performs better on the given dataset. The confusion matrix shows how many predictions were correct and how many were incorrect. The diagonal values in the confusion matrix represent correct predictions, while the other values represent misclassifications.

The Flask-based web application demonstrated fast response time, enabling predictions to be generated within a few seconds after user input. Since the trained model is loaded and used through the Flask backend, predictions are generated quickly after user input is submitted. The dashboard and forms are user-friendly, allowing users to navigate easily through the application.

Overall, the experimental results indicate that the proposed system is reliable for educational disease prediction, healthcare awareness, and academic demonstration

3.3 Discussion

The proposed system demonstrates the practical use of machine learning in healthcare prediction. One of the main advantages of the system is that it combines two prediction approaches in a single application. The symptom-based module predicts diseases using selected symptoms, while the health risk assessment module analyses numerical health values to provide risk-based output.

The system is useful from an educational point of view because it shows the complete workflow of a machine learning web application. It includes dataset preprocessing, feature selection, model training, model saving, Flask integration, frontend design, prediction result display, and PDF report generation. This makes the project suitable for students who want to understand both machine learning and web development.

The application provides a modern, responsive, and user-friendly interface that enables users to navigate easily between different modules. Users can access prediction modules through the dashboard, enter details easily, and view results clearly. The PDF report feature improves the usefulness of the system because users can save prediction results for reference. The prediction history feature enhances usability by allowing users to review previously generated predictions

Module	Function
Login	User authentication
Dashboard	Navigation and overview
Symptom Prediction	Predict disease using symptoms
Health Risk Assessment	Assess risk from health parameters
Prediction History	View previous predictions
PDF Report	Download prediction report
Profile	Manage user information

Despite its advantages, the proposed system has certain limitations. The prediction accuracy depends on the dataset used for training. If the dataset is small or incomplete, the result may not be highly accurate. The system can predict only the diseases available in the dataset. It is intended only as a decision-support tool and should not replace professional medical consultation, laboratory investigations, or clinical diagnosis.

The system is mainly designed for educational and awareness purposes. It can encourage users to take symptoms seriously and consult a doctor when required. Therefore, the system includes a medical disclaimer to avoid misunderstanding. This makes the application responsible while demonstrating the role of machine learning in healthcare.

4. CONCLUSION

4.1 Summary

This research presented an **Machine Learning Based Disease Prediction System using Python**, developed using the Flask framework to provide an interactive healthcare prediction platform. The system consists of two primary modules: **Symptom-Based Disease Prediction** and **Health Risk Assessment**. Machine learning algorithms were trained using healthcare datasets, and the trained models were integrated into a web application to generate predictions based on user inputs. Additional features such as a responsive dashboard, prediction history, PDF report generation, health recommendations, and input validation improve the usability and overall effectiveness of the system.

4.2 Key Findings

- The proposed system successfully integrates **Machine Learning** with the **Flask** web framework for disease prediction.
- Two prediction modules - **Symptom-Based Disease Prediction** and **Health Risk Assessment** provide healthcare insights based on different user inputs.
- Among the evaluated algorithms, **Random Forest** achieved the highest prediction accuracy and was selected as the primary model.
- The web application provides a **responsive and user-friendly interface** that simplifies interaction for users.
- Features such as **prediction history**, **PDF report generation**, and **health recommendations** enhance the practicality of the application.
- The system demonstrates that machine learning can effectively assist in preliminary healthcare decision support.

4.3 Applications

The proposed system can be applied in various healthcare and educational environments, including:

- **Hospitals** – To support preliminary disease prediction and assist healthcare professionals.
- **Clinics** – To perform quick initial health assessments before detailed diagnosis.
- **Educational Institutions** – As a learning tool for students studying Machine Learning, Data Science, and Healthcare Informatics.
- **Healthcare Awareness Programs** – To educate individuals about health risks and encourage early medical consultation based on predictive analysis.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 7| July 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

4.4 Future Work

The proposed system can be further enhanced through several improvements:

- Incorporating **Deep Learning** techniques to improve prediction accuracy for complex diseases.
- Integrating **IoT-based wearable devices** for real-time health monitoring and automatic data collection.
- Deploying the application on **cloud platforms** to enable secure remote access and improved scalability.
- Providing **doctor consultation and appointment scheduling** features for seamless healthcare support.
- Developing a dedicated **mobile application** to make the system more accessible across Android and iOS devices.

REFERENCES

1. *Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.*
2. *Biau, G., & Scornet, E. (2016). A random forest guided tour. TEST, 25(2), 197–227.*
3. *Kaur, H., & Kumari, V. (2020). Predictive modelling and analytics for diabetes using machine learning approach. Applied Computing and Informatics, 18(1/2), 90–100.*
4. *Chen, M., Hao, Y., Hwang, K., Wang, L., & Wang, L. (2017). Disease prediction by machine learning over big data from healthcare communities. IEEE Access, 5, 8869–8879.*
5. *Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. The New England Journal of Medicine, 380(14), 1347–1358.*
6. *Deo, R. C. (2015). Machine learning in medicine. Circulation, 132(20), 1920–1930.*
7. *Sidey-Gibbons, J. A. M., & Sidey-Gibbons, C. J. (2019). Machine learning in medicine: A practical introduction. BMC Medical Research Methodology, 19, 64.*
8. *Dey, A. (2016). Machine learning algorithms: A review. International Journal of Computer Science and Information Technologies, 7(3), 1174–1179.*
9. *Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. BMC Medical Informatics and Decision Making, 19, 281.*
10. *Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep EHR: A survey of recent advances in deep learning techniques for electronic health record analysis. IEEE Journal of Biomedical and Health Informatics, 22(5), 1589–1604.*

Web/Tool References

11. *Flask Documentation. (n.d.). Flask web development framework. Retrieved from https://flask.palletsprojects.com/*
12. *Scikit-learn Developers. (n.d.). Scikit-learn: Machine learning in Python. Retrieved from https://scikit-learn.org/*
13. *SQLite Documentation. (n.d.). SQLite database engine. Retrieved from https://www.sqlite.org/*
14. *ReportLab Documentation. (n.d.). ReportLab PDF generation toolkit. Retrieved from https://www.reportlab.com/*
15. *Python Software Foundation. (n.d.). Python documentation. Retrieved from https://docs.python.org/3/*