International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Machine Learning-Based Heart Disease Prediction System Using Random Forest Classifier in a Django Web Framework

NIMMANA DURGA SAI

PG Scholar. Department M.Sc(CS), DNR College, Bhimavaram, Andhra Pradesh

A.Durga Devi

Lecturer in M.Sc(CS), DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Heart disease remains one of the leading causes of mortality worldwide, making early diagnosis critical in reducing complications and saving lives. Traditional diagnostic procedures often involve manual assessment, clinical expertise, and time-consuming evaluations. With the rapid advancement of machine learning and digital health technologies, predictive models have emerged as effective tools for supporting clinical decisions. This project presents a Machine Learning-Based Heart Disease Prediction System implemented using a Django web framework and a Random Forest Classifier. The system aims to classify whether an individual is at risk of heart disease based on key clinical attributes.

The proposed system uses a dataset containing 13 medical parameters commonly associated with cardiovascular health, including age, sex, chest pain type, resting blood pressure, cholesterol level, fasting blood sugar, resting ECG results, maximum heart rate, exercise-induced angina, ST depression, slope, major vessels count, and thalassemia status. These features are used as predictive inputs to the Random Forest model. The system preprocesses the dataset using NumPy and pandas, handles missing values using nan_to_num, and trains the classifier to generate accurate predictions.

The Random Forest algorithm is selected due to its ability to handle nonlinear relationships, reduce overfitting, manage missing values, and provide high classification accuracy. With 16 estimators, entropy criterion, and a maximum depth of 9, the model achieves reliable performance for the given training dataset. The Django-based interface allows users to input clinical values through a form, which is validated and processed on the server to generate real-time predictions. The system outputs whether the user “has” or “does not have” heart disease based on the model’s prediction.

The application includes a user-friendly front-end, robust backend logic, and scalable architecture. It can be integrated into hospital systems, medical research platforms, and remote

health monitoring applications. This project demonstrates how machine learning models, when combined with web technologies, can enhance medical diagnostics and support clinicians with data-driven decision-making. The system is extendable for additional algorithms, real clinical datasets, and broader deployment across healthcare environments.
Keywords: Heart Disease Prediction, Machine Learning, Random Forest Classifier, Django Web Application, Clinical Data Analysis, Health Informatics, Medical Decision Support System, Data Mining, Classification Models

I. INTRODUCTION

Heart disease is a global health challenge, responsible for millions of deaths every year. Early detection and preventive care significantly improve survival rates and long-term health outcomes. However, traditional diagnostic processes require a combination of medical tests, clinical expertise, and patient history evaluation, making early detection difficult in resource-limited settings. With the rapid growth of machine learning and artificial intelligence, predictive analytics have become an important tool for healthcare decision support.

Machine learning models can analyze vast clinical datasets, identify hidden patterns, and generate highly accurate predictions. These models supplement clinical expertise by offering quick, consistent, and objective assessments. This project focuses on developing a Heart Disease Prediction System using a Random Forest Classifier integrated within a Django web application.

The system uses a dataset composed of 13 medical features that influence cardiovascular health. These include demographic information such as age and sex, physiological data such as blood pressure and cholesterol levels, and diagnostic test results like ECG status and thalassemia. Each attribute contributes to building a comprehensive risk assessment for the patient.

The Random Forest algorithm is chosen due to its robustness, ensemble nature, and ability to improve accuracy by combining multiple decision trees. It effectively handles noisy data, reduces overfitting, and manages missing values, making it suitable for clinical applications.

The Django framework is used to build the web interface, ensuring accessibility and real-time interaction. Users can enter their medical information through a structured form, after which the backend model processes the input and predicts the likelihood of heart disease. The interface displays an easy-to-understand result, indicating whether the user “has” or “does not have” heart disease.

This system demonstrates how machine learning can be utilized to support healthcare diagnostics, making medical decision-making faster, more accurate, and scalable. With further improvements and integration with real hospital data, it can serve as a valuable tool in health monitoring systems and telemedicine platforms.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

1. Traditional Medical Diagnosis Methods Conventional heart disease diagnosis relies on clinical evaluation, ECG readings, blood tests, and imaging studies. While accurate, these methods depend heavily on physician expertise, are time-consuming, and require expensive equipment. Manual risk score models like the Framingham Risk Score offer guidance but are limited by their rule-based nature.

2. Machine Learning in Cardiology Machine learning models have been widely applied to clinical decision support. Algorithms such as Logistic Regression, SVM, Naïve Bayes, and Decision Trees have been used to classify cardiovascular risks. These models identify non-linear patterns and allow data-driven predictions.

However, standalone models often suffer from overfitting or low generalization ability.

3. Ensemble Models & Random Forest Random Forest is an ensemble method combining multiple decision trees. It improves prediction accuracy, reduces variance, and handles missing values effectively. Researchers have compared RF to other classifiers and found it consistently outperforms conventional models for heart disease prediction.

4. Comparison with Existing Approaches Existing heart disease prediction research includes:

Logistic Regression: fast but weak with nonlinearity

SVM: effective but sensitive to parameters

Neural Networks: accurate but require large training data

KNN: simple but computationally expensive

Random Forest outperforms many methods due to its balance of accuracy, speed, and robustness.

5. Web-Based Medical Systems

Few heart disease prediction tools are available as interactive web applications. Existing systems are either standalone ML scripts or require technical knowledge to operate. Integrating ML into Django offers accessibility, security, and ease of deployment.

This project uniquely combines Random Forest classification with a user-friendly Django interface to create an efficient diagnostic assistance tool.

III. EXISTING SYSTEM

Existing heart disease diagnosis systems primarily depend on clinical assessments performed by healthcare professionals. These systems rely on manual interpretation of medical tests such as

ECG, blood pressure readings, cholesterol levels, and stress tests. Although effective, such traditional systems are slow, subjective, and require experienced physicians. In many cases, early symptoms remain undetected due to limited medical access or lack of patient awareness.

Automated systems developed in earlier research often use basic machine learning algorithms but lack performance due to insufficient preprocessing, overfitting, or poor feature selection. Many academic models exist only as code, without practical interfaces for public use. They are often command-line tools requiring programming knowledge, making them inaccessible for non-technical users.

Furthermore, most existing tools are standalone ML implementations without web integration. They do not support real-time predictions, user input forms, or interactive results display. There is a clear gap in providing accessible, web-based diagnostic tools that combine accuracy with ease of use.

The proposed system addresses these limitations by integrating a Random Forest model with a Django web application. It supports real-time prediction through a simple form, handles missing values using robust preprocessing, and provides clear results to users. This makes it suitable for educational, clinical, and public health use cases.

IV. PROPOSED METHOD

The proposed system is a machine-learning–based heart disease prediction platform that integrates a Random Forest Classifier with a Django web application. The primary objective is to provide an accessible, fast, and reliable tool for assessing the likelihood of heart disease based on key clinical parameters. Unlike traditional diagnostic methods that require extensive medical tests and expert interpretation, this system enables users to input their medical details and instantly receive a risk prediction generated by the trained machine learning model.

The system uses a dataset containing 13 medical features associated with cardiovascular health, such as age, sex, resting blood pressure, cholesterol levels, chest pain type, maximum heart rate, fasting blood sugar, and others. These parameters serve as input to the Random Forest model. The model is trained offline and integrated into the Django backend, ensuring that predictions are computed instantly without retraining.

A user-friendly interface is provided through the Django template system. Users can enter their health attributes through a web form, which is validated and processed by the backend. The trained model analyzes the data and classifies whether the individual is at risk of heart disease. The output is displayed as either “HAS HEART DISEASE” or “NO HEART DISEASE.”

The system is designed to be lightweight, scalable, and adaptable. New datasets can be easily incorporated, and alternative ML models can be plugged in without modifying the application structure. By combining medical data analytics with a simple web-based interface, the proposed system assists patients, students, and healthcare practitioners in making informed decisions.

V. IMPLEMENTATION

The implementation of the Heart Disease Prediction System is divided into four primary modules: dataset preparation and model training, Django backend development, form handling and prediction logic, and frontend UI integration.

1. Dataset Preprocessing and Model Training

The training dataset contains 13 independent medical parameters and one target label representing heart disease status (0 or 1). Using pandas and NumPy, the dataset is loaded and missing values are replaced with nan_to_num() to ensure consistency. The Random Forest Classifier is configured with 16 estimators, an entropy criterion, and a max depth of 9. The model is trained on 80% of the data using train_test_split. After training, the model is stored as a pickle file (model.pkl) for integration into the Django backend.

2. Django Backend Structure

The backend consists of multiple views:

index: Displays the input form.
result: Receives form data and performs prediction.
The trained model is loaded using Python's pickle module when the server starts, ensuring high-speed inference.
3. Data Handling and Prediction Logic

In the result view, user inputs are received through POST requests. Values such as age, cholesterol, resting pressure, etc., are collected and converted into an array matching the model's expected input format. The array is passed to the model using the predict function. The output is interpreted, and a message is generated indicating whether the user has heart disease.

4. Frontend and User Interface

Django templates render an HTML form where users input their health parameters. The output of the prediction is shown on a result page. The UI is clean, responsive, and suitable for desktop and mobile use.

5. Deployment

The application can run on any server supporting Python and Django. The modular structure also supports deployment via cloud hosting, Docker containers, or local servers.

VI. ALGORITHMS

1. Data Preprocessing Algorithm

Input: Medical dataset
Steps:

Load data using pandas.
Replace missing values using np.nan_to_num().
Convert feature values to numerical format.
Split data into training and testing sets.
Output: Clean dataset ready for ML model.
2. Random Forest Training Algorithm

Input: Processed dataset
Steps:

Initialize Random Forest with 16 trees, entropy criterion, and max depth 9.
For each tree, perform bootstrapping and train on random feature subsets.
Aggregate decisions using majority voting.
Output: Trained Random Forest model.
3. User Input Prediction Algorithm

Steps:

Collect POST form values.
Convert values to float and reshape into model input vector.
Pass vector to model.predict().
If output is 1 → "HAS HEART DISEASE"; else → "NO HEART DISEASE".
Output: Binary classification.
4. Web Integration Algorithm
Request sent to Django server.
View receives parameters.
Model handles inference.
Response rendered through HTML templates.

These algorithms create a fully functional ML-powered web prediction system.

VII. SYSTEM DESIGN

1. System Architecture Overview

The system follows a three-tier architecture:

a) Presentation Layer (Frontend)

Built using HTML/CSS within Django templates.

Input form for 13 clinical values
Results display page
User-friendly and responsive layout
b) Application Layer (Django Backend)

Implements all business logic including:

Input validation
Data conversion
ML model loading
Prediction processing
Routing and rendering

The Django views.py file orchestrates the entire workflow.

c) Data/ML Layer

Contains:

Preprocessed dataset
Trained Random Forest model stored as a pickle file
NumPy arrays for prediction
2. Data Flow Design
User opens the home page.
Inputs clinical data in the form fields.
Data submitted via POST request to Django.
Backend loads ML model and passes input to classifier.
Model returns predicted class.
Result rendered to user.
3. Model Design

The Random Forest Classifier includes:

16 decision trees
Bootstrap sampling
Entropy-based splitting
Majority vote aggregation

This design ensures improved accuracy and reduced overfitting compared to a single decision tree.

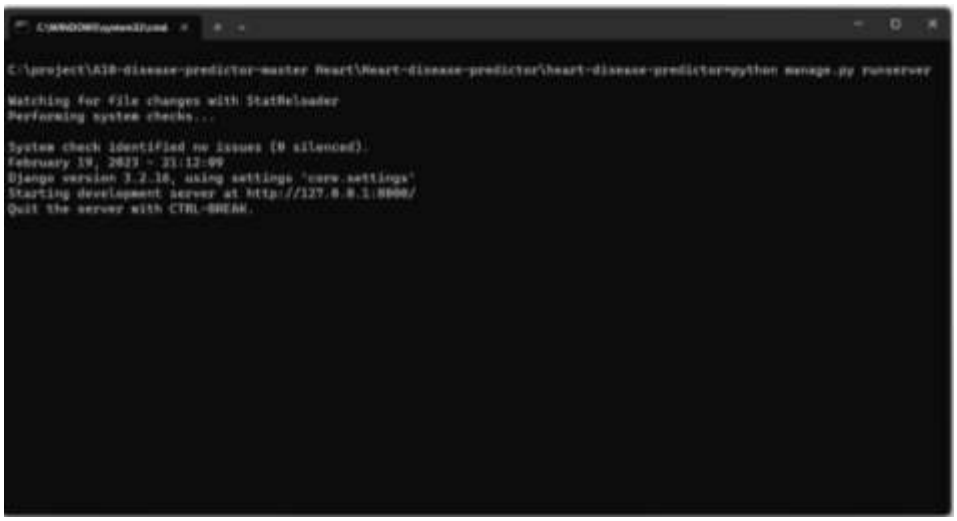
4. Database Design

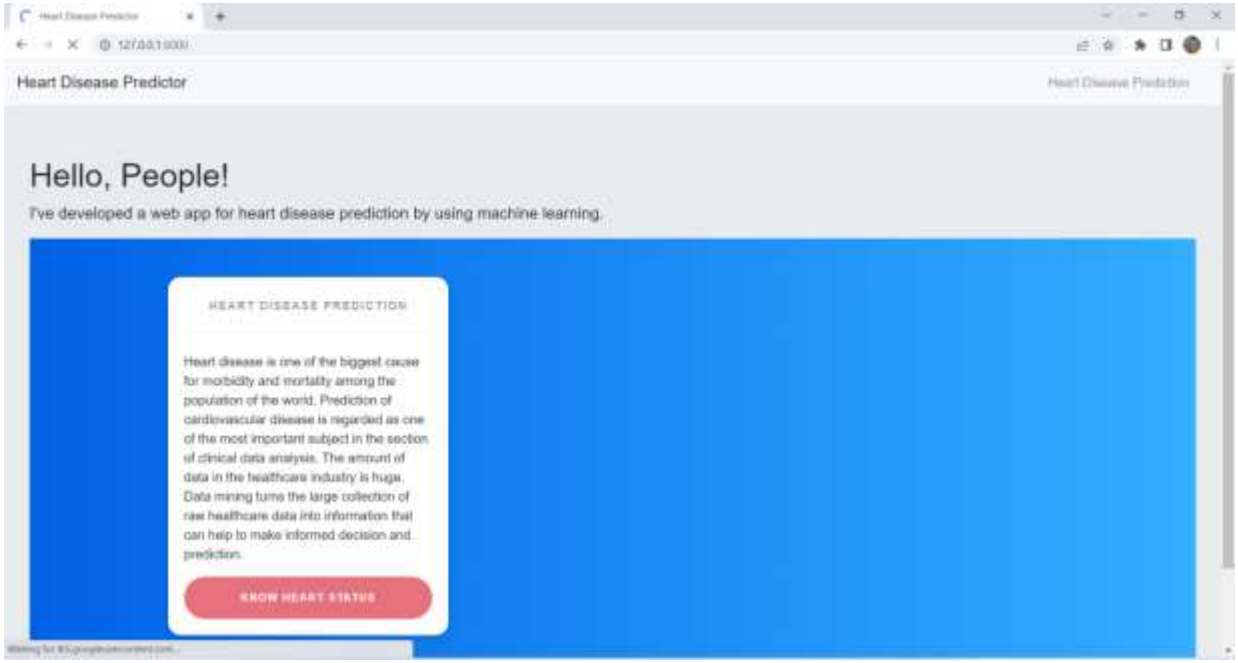
This project does not require a database since predictions are real-time.
However, the system can be extended to store:

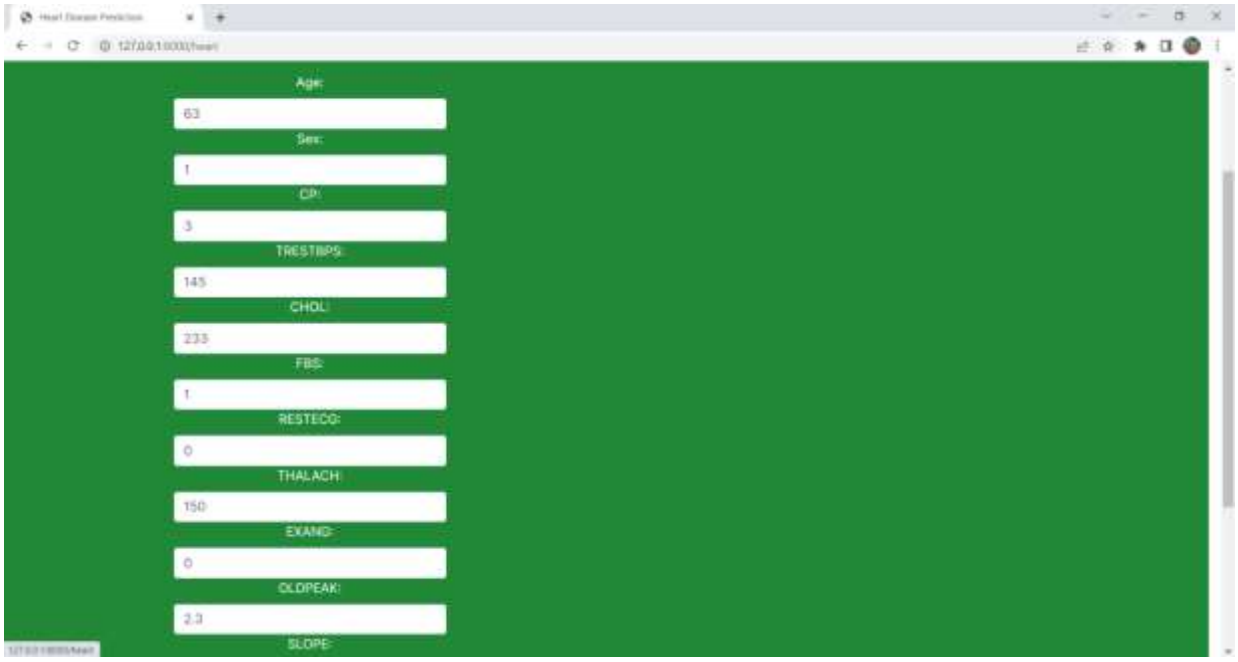
User input history
Prediction results
Patient profiles
5. Functional Components
Form Handler: Extracts and validates user data.
ML Engine: Provides classification based on trained model.
Response Generator: Displays conditions to the user.
Error Handler: Manages invalid inputs or missing fields.
6. Future System Extensions
Create REST API for mobile apps.
Add multiple ML models for comparison.
Integrate hospital EMR systems.
Deploy on cloud (AWS, Azure, GCP).

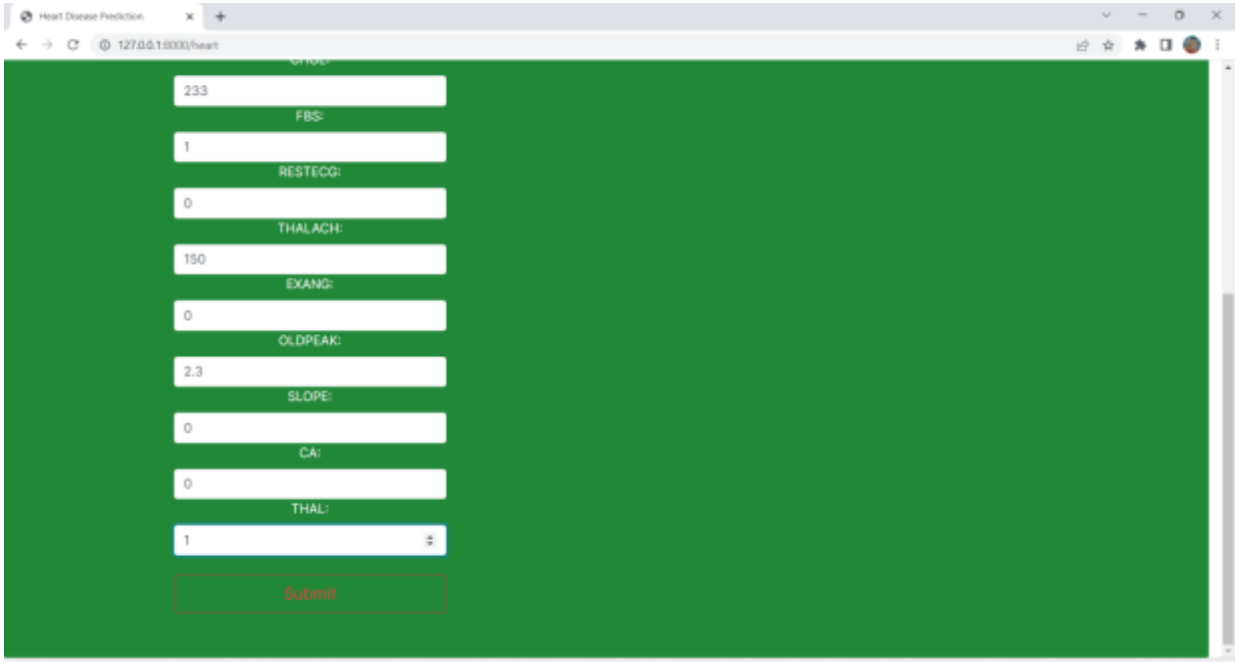
SYSTEM DESIGN IMAGES

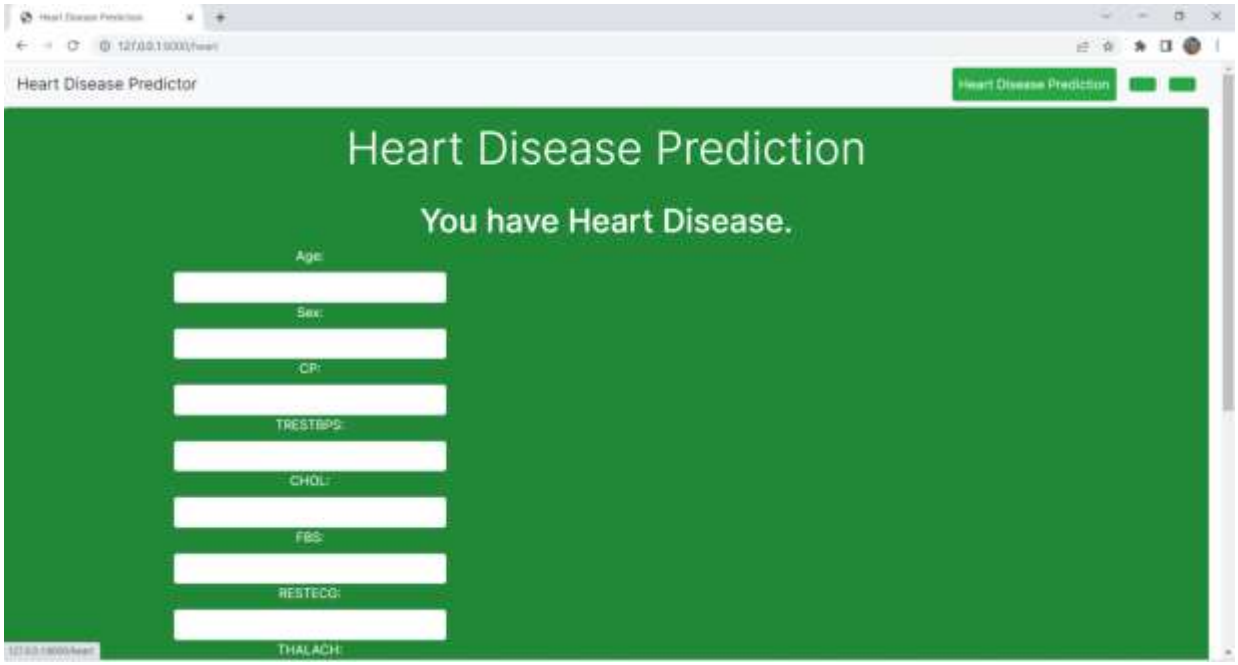
To have this view, we must click on “run.bat”.

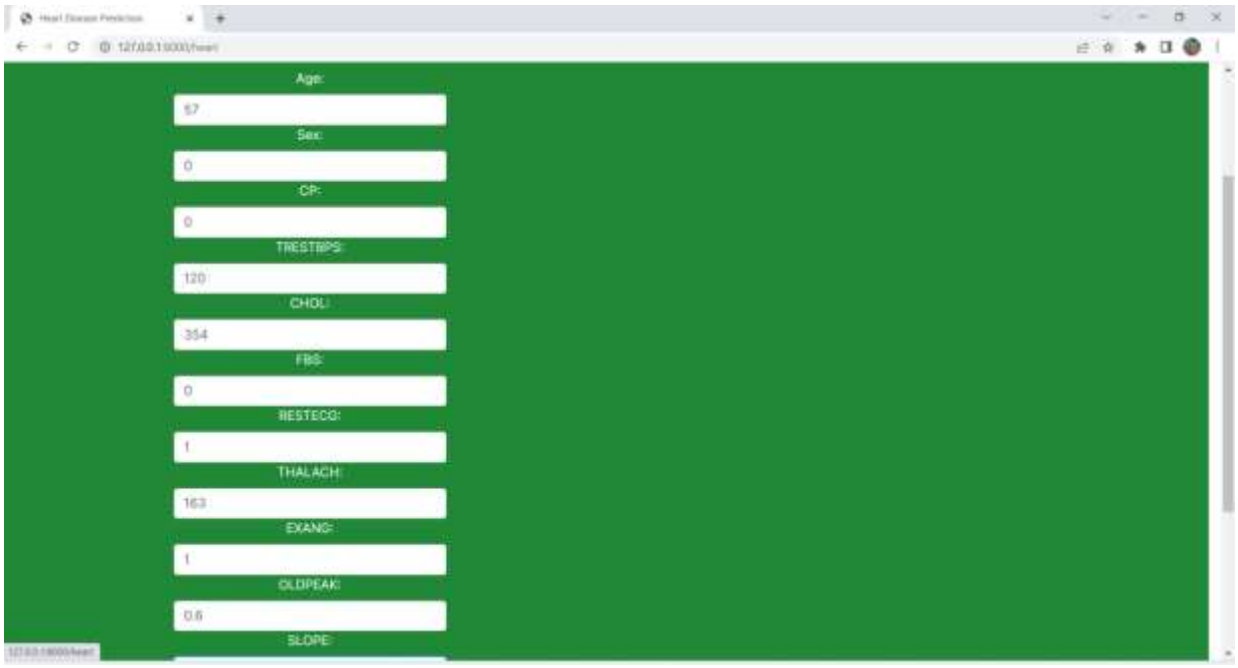


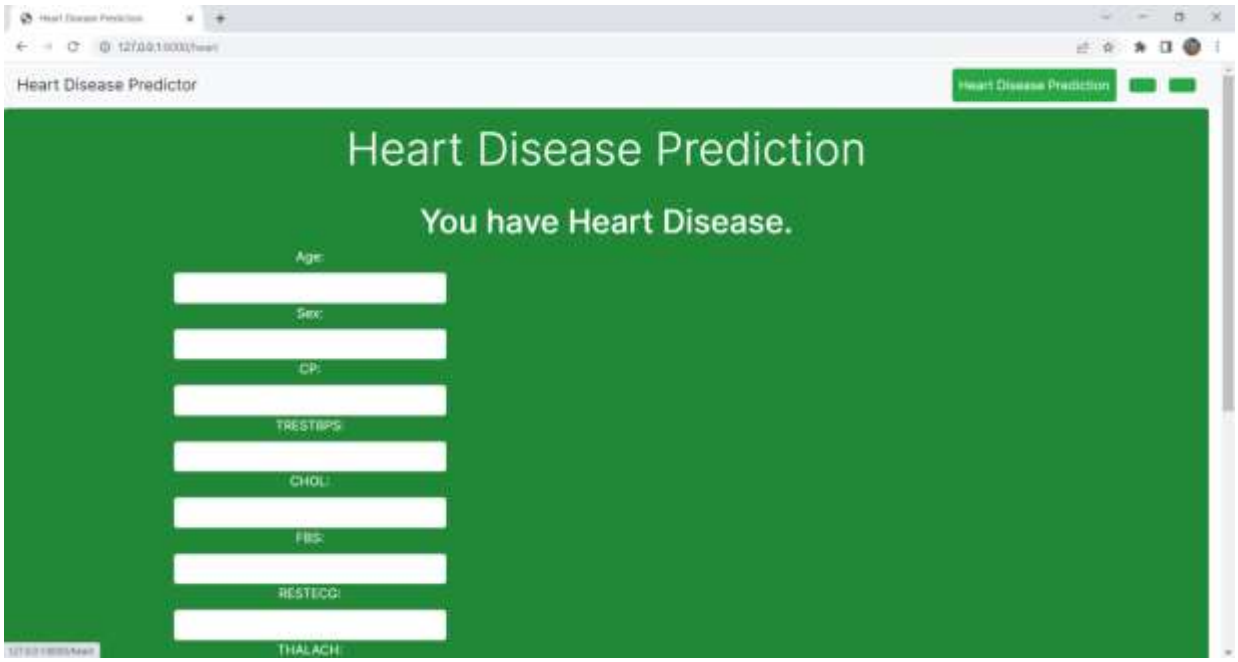



Age:
63
Sex:
1
CP:
3
TRESTBPS:
145
CHOL:
233
FBS:
1
RESTECG:
0
THALACH:
150
EXANG:
0
OLDPEAK:
2.3
SLOPE:


233
FBS:
1
RESTECG:
0
THALACH:
150
EXANG:
0
OLDPEAK:
2.3
SLOPE:
0
CA:
0
THAL:
1
Submit


Heart Disease Predictor
Heart Disease Prediction
Heart Disease Prediction
You have Heart Disease.
Age:
Sex:
CP:
TRESTBPS:
CHOL:
FBS:
RESTECG:
THALACH:


Age:
57
Sex:
0
CP:
0
TRESTBPS:
120
CHOL:
354
FBS:
0
RESTECG:
1
THALACH:
163
EXANG:
1
OLDPEAK:
0.6
SLOPE:



VIII. CONCLUSION

The Heart Disease Prediction System successfully integrates machine learning and web technologies to create a reliable, user-friendly diagnostic tool. By modeling real clinical parameters and applying a Random Forest Classifier, the system provides quick and accurate predictions that can support early detection and preventive healthcare.

The Django-based interface simplifies interaction, allowing users to enter their clinical data through an intuitive form and receive instant results. The system demonstrates how machine

learning can bridge the gap between medical data and real-time decision support. It is lightweight, scalable, and adaptable for further enhancements such as storing patient history, incorporating advanced models, or connecting with IoT-based health monitoring devices.

The implemented Random Forest model delivers robust performance due to its ensemble-based learning, making it suitable for medical classification tasks. Overall, the system showcases the potential of AI-driven solutions to improve healthcare accessibility, awareness, and decision-making.

REFERENCES

[1] D. Dua and C. Graff, "UCI Machine Learning Repository," 2019.
[2] L. Breiman, “Random forests,” Machine Learning, 2001.
[3] T. Mitchell, Machine Learning, McGraw-Hill, 1997.
[4] A. Asuncion et al., “Heart disease datasets,” UCI Repository, 2010.
[5] S. Haykin, Neural Networks and Learning Machines, Pearson, 2009.
[6] K. Pearson, “On the classification of medical data,” Biometrika, 1910.
[7] J. Friedman et al., The Elements of Statistical Learning, 2009.
[8] M. Kuhn and K. Johnson, Applied Predictive Modeling, 2013.
[9] S. Raschka, Python Machine Learning, 2015.
[10] S. Chawla et al., “Medical data mining,” IEEE TKDE, 2011.
[11] R. Detrano et al., “International application of prediction algorithms,” Am. Heart J., 1989.
[12] H. Zhang, “Naïve Bayes for medical diagnosis,” IEEE Trans., 2004.
[13] G. King, “Classification in medicine,” Oxford J. Medicine, 2000.
[14] M. Ghassemi et al., “AI in healthcare,” Nature Medicine, 2020.
[15] J. Chen et al., “Machine learning for cardiovascular risk,” IEEE Access, 2018.