Title:  **Predictive Pulse: Harnessing Machine Learning for Blood Pressure Analysis**

Team Member : Zobiya Sayyed, Janhavi Phale, Minakshee Bhagat.

Technology: Python, Flask, HTML/CSS, Pandas, Scikit-learn, Matplotlib, Seaborn

1. Introduction

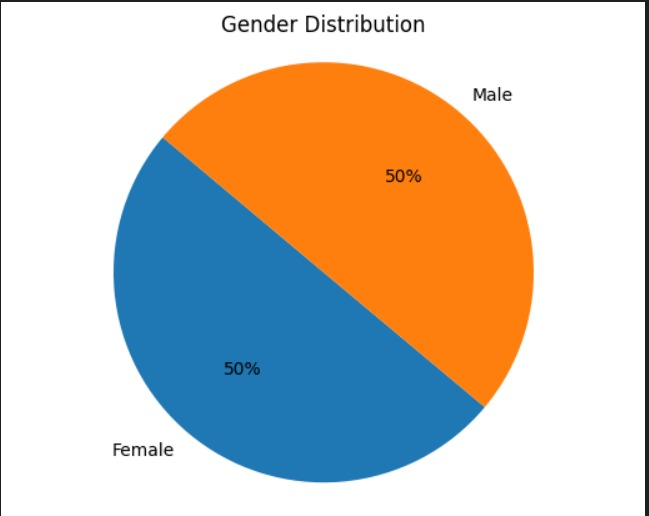
* The Blood Pressure Prediction System is a web-based application that predicts the stage of hypertension based on patient data such as age, gender, systolic and diastolic pressure, lifestyle factors, and medical history.
* The project leverages machine learning models trained on patient datasets to make accurate predictions and provides visual analytics to understand the dataset trends.

1. Objective

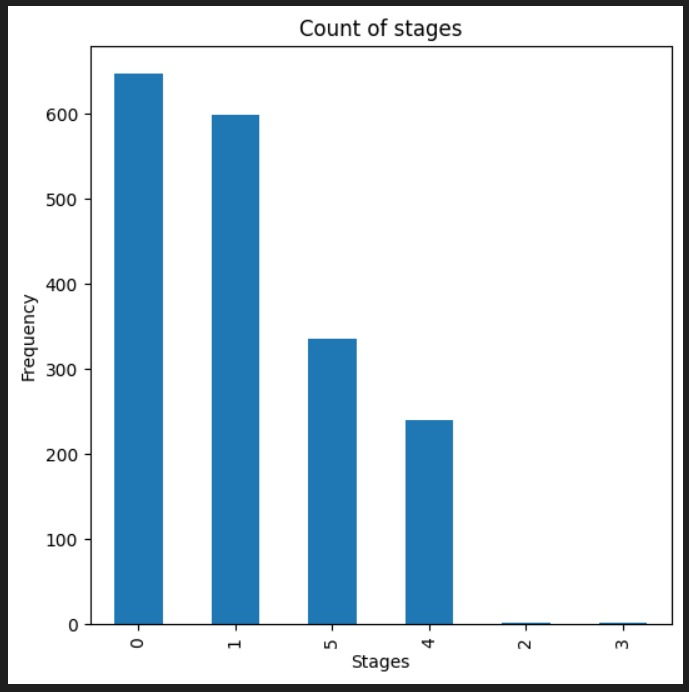
* To design and develop a predictive system for classifying hypertension stages.
* To provide an interactive web interface for both data analysis and predictions.
* To integrate data visualization for better insight into the dataset.
* To store and reuse a trained ML model for real-time predictions.

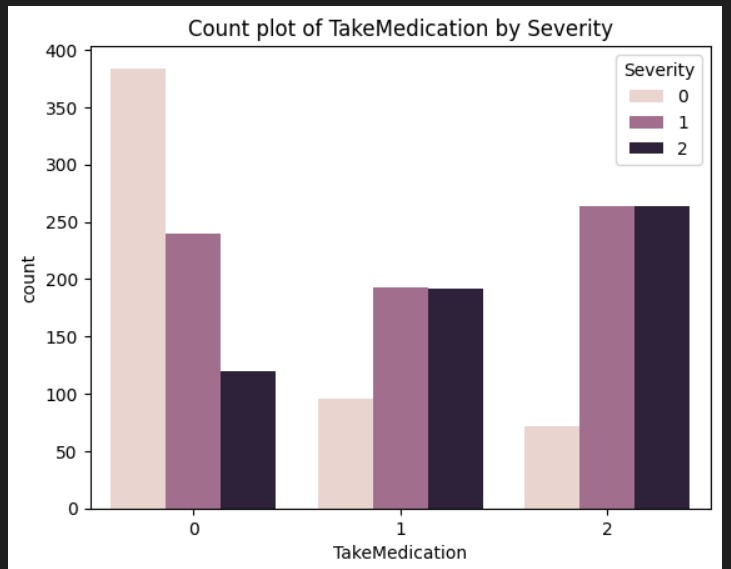
1. Features

* Dataset Analysis
* Displays the first few rows of the dataset.
* Shows key statistics like gender distribution, stage counts, and medication severity.
* Data Visualization
* Pie chart for gender distribution.

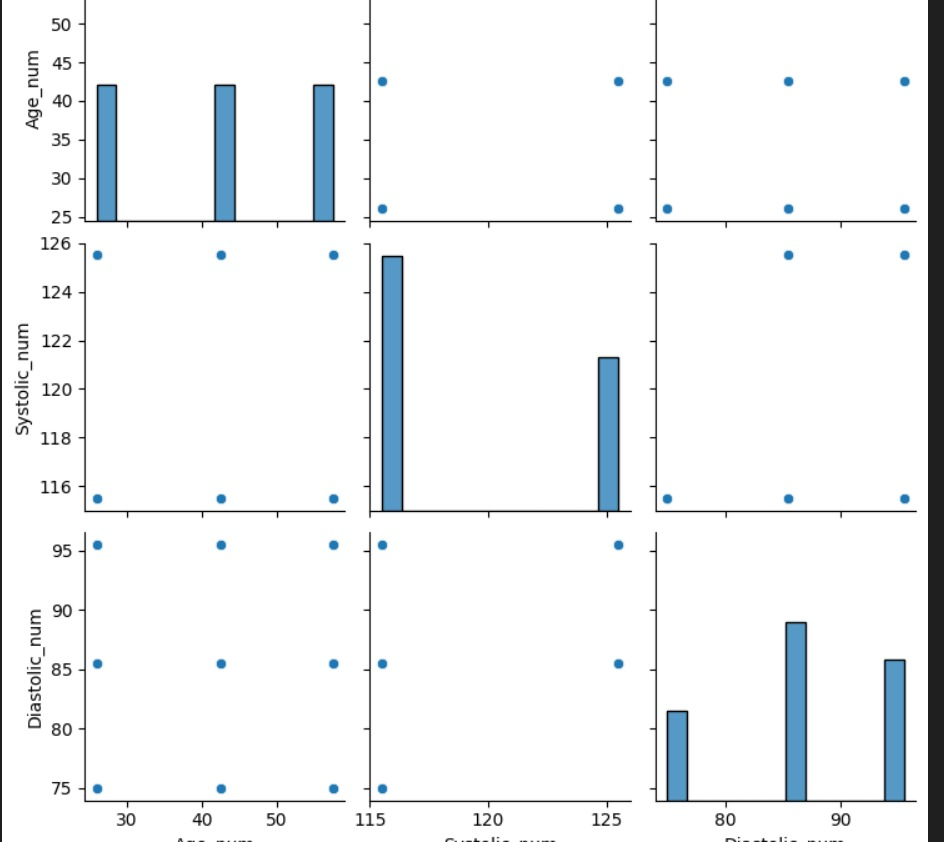


* Bar chart for hypertension stages.





* Pair plots for numeric features.



* Machine Learning Model
* Multiple algorithms tested: Logistic Regression, Decision Tree, Random Forest, Naive Bayes.
* Model with the best accuracy saved as model.pkl.
* Web Application
* / route: Analysis dashboard with dataset preview and charts.
* /details route: Form to enter patient details.
* /predict route: Outputs the hypertension stage.

1. System Architecture

Flow:

* Data Preprocessing
* Handling missing values.
* Label encoding categorical features.
* Cleaning incorrect labels.
* Model Training
* Splitting the dataset into training and testing sets.
* Testing different ML classifiers.
* Selecting the best model based on accuracy score.
* Model Deployment
* Saving trained model to model.pkl.
* Loading the model in Flask for predictions.
* Frontend
* HTML templates for dashboard, input form, and result page.
* Static folder for images and CSS.

1. Technology Stack

* Backend: Python 3.13, Flask
* Frontend: HTML, CSS, Bootstrap
* Data Analysis & ML: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
* Model Storage: Pickle (model.pkl)

1. File Structure

project/

│

├── app.py # Flask application

├── A.py # Data analysis & model training

├── model.pkl # Saved ML model

├── patient\_data.csv # Dataset

├── templates/

│ ├── index.html # Dashboard

│ ├── details.html # Input form

│ ├── result.html # Prediction result

1. How to Run

Install dependencies

pip install flask pandas numpy scikit-learn matplotlib seaborn

Train the model (if not already done)

Run A.py to preprocess the dataset, train, and save model.pkl.

Run the Flask app

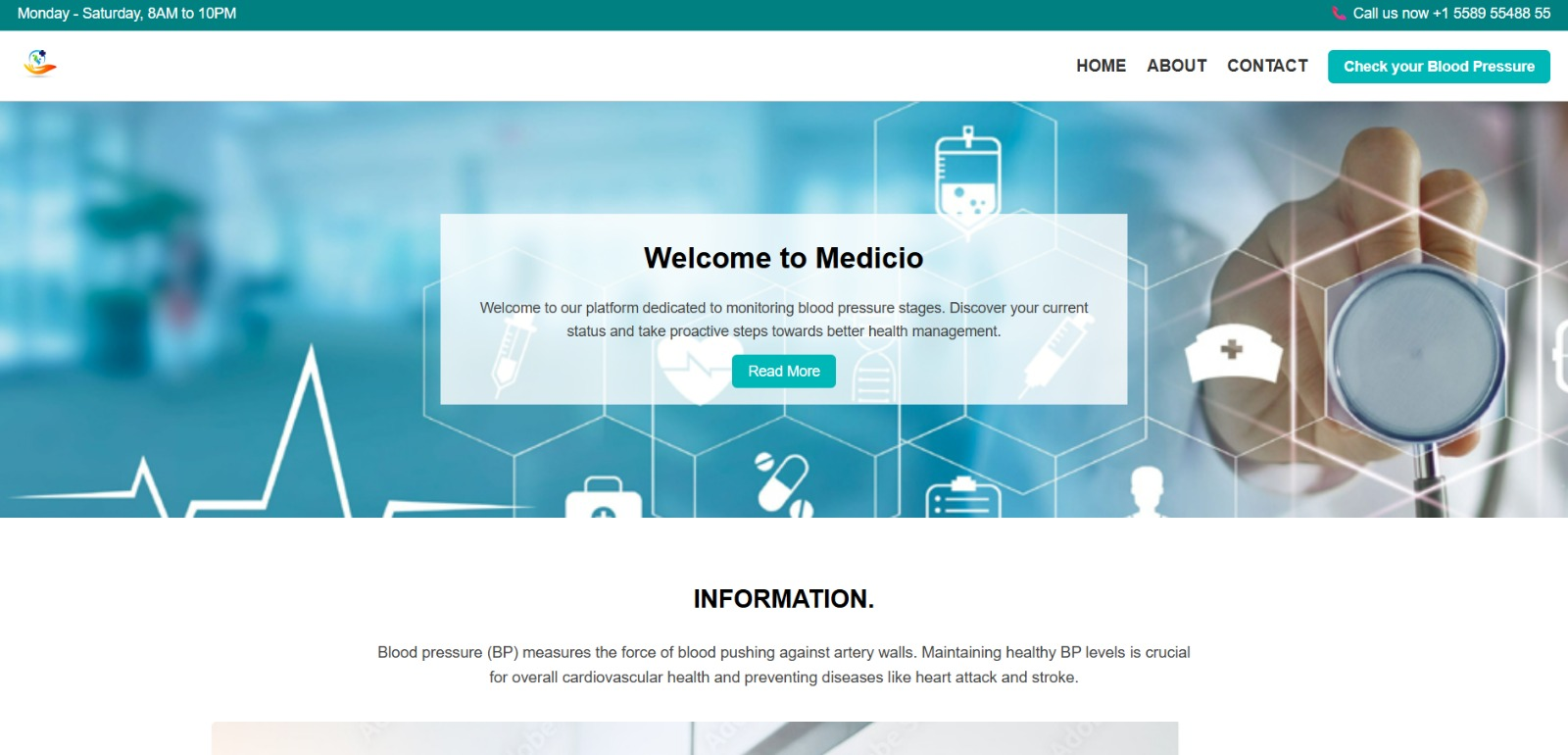
python app.py

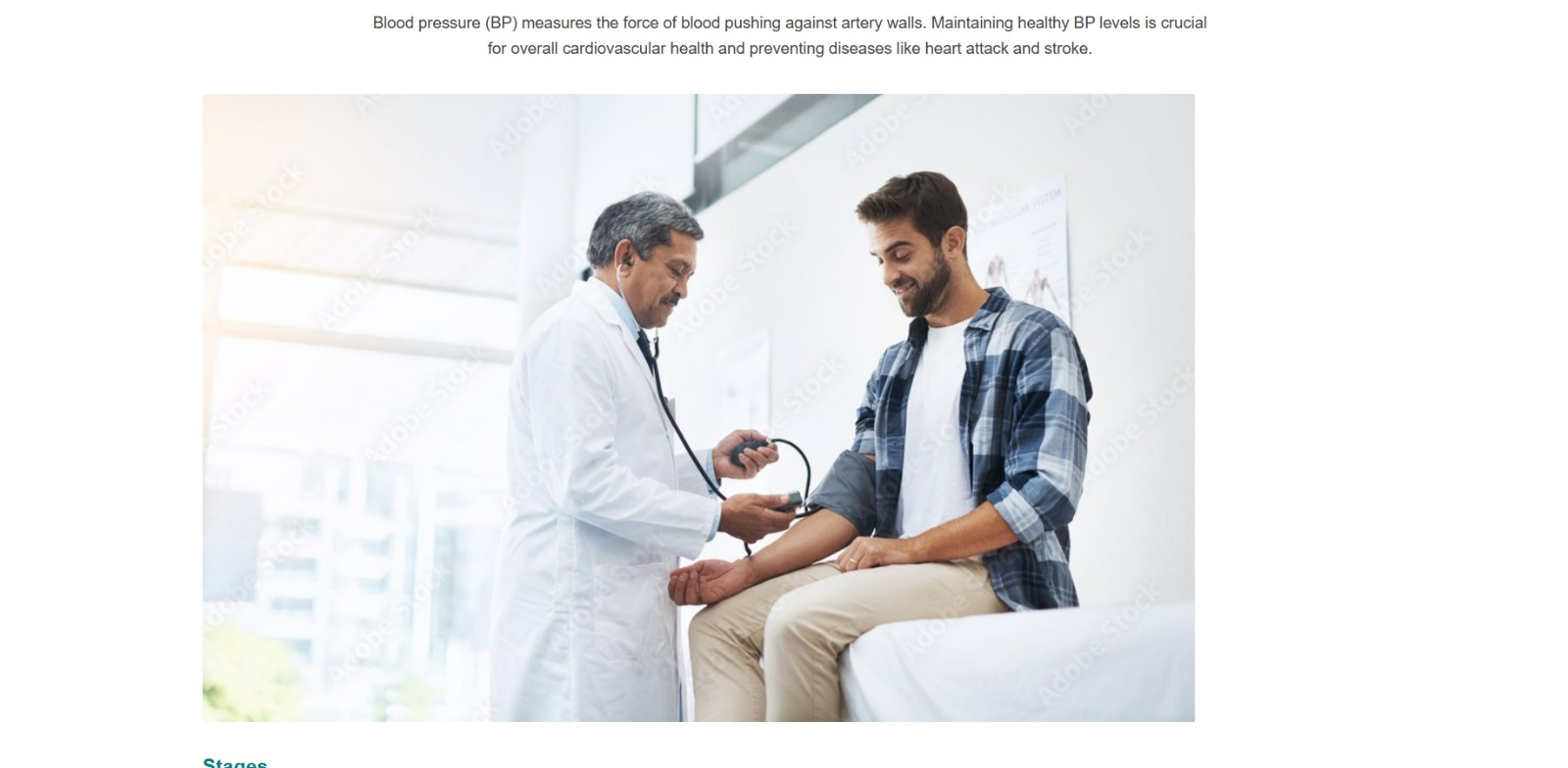
Access in browser

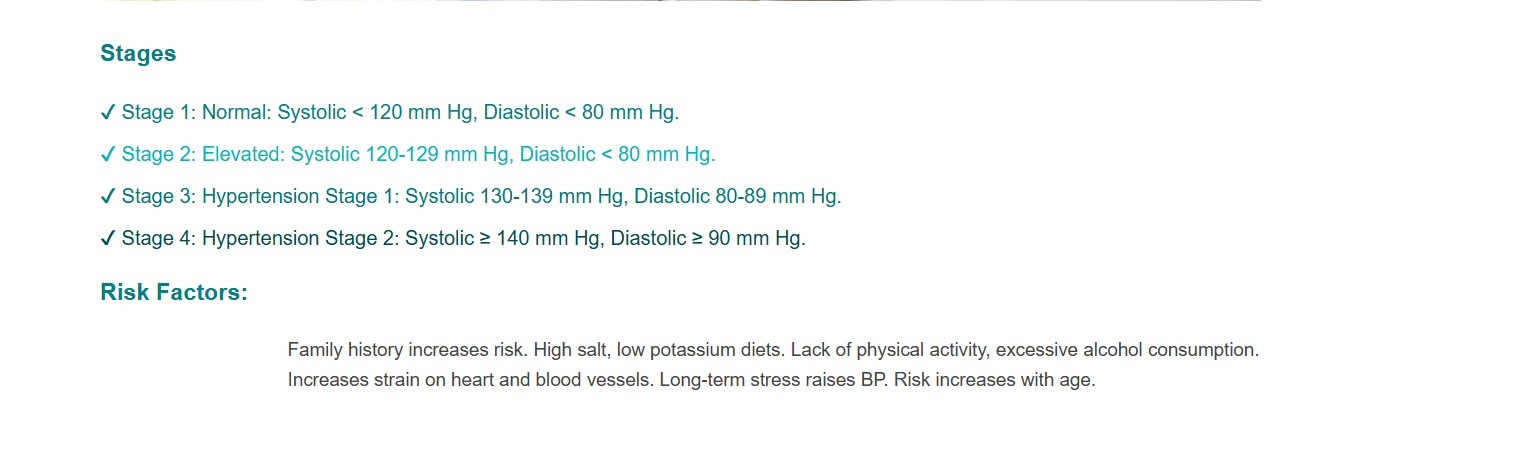
Open http://127.0.0.1:5000

1. Screenshots

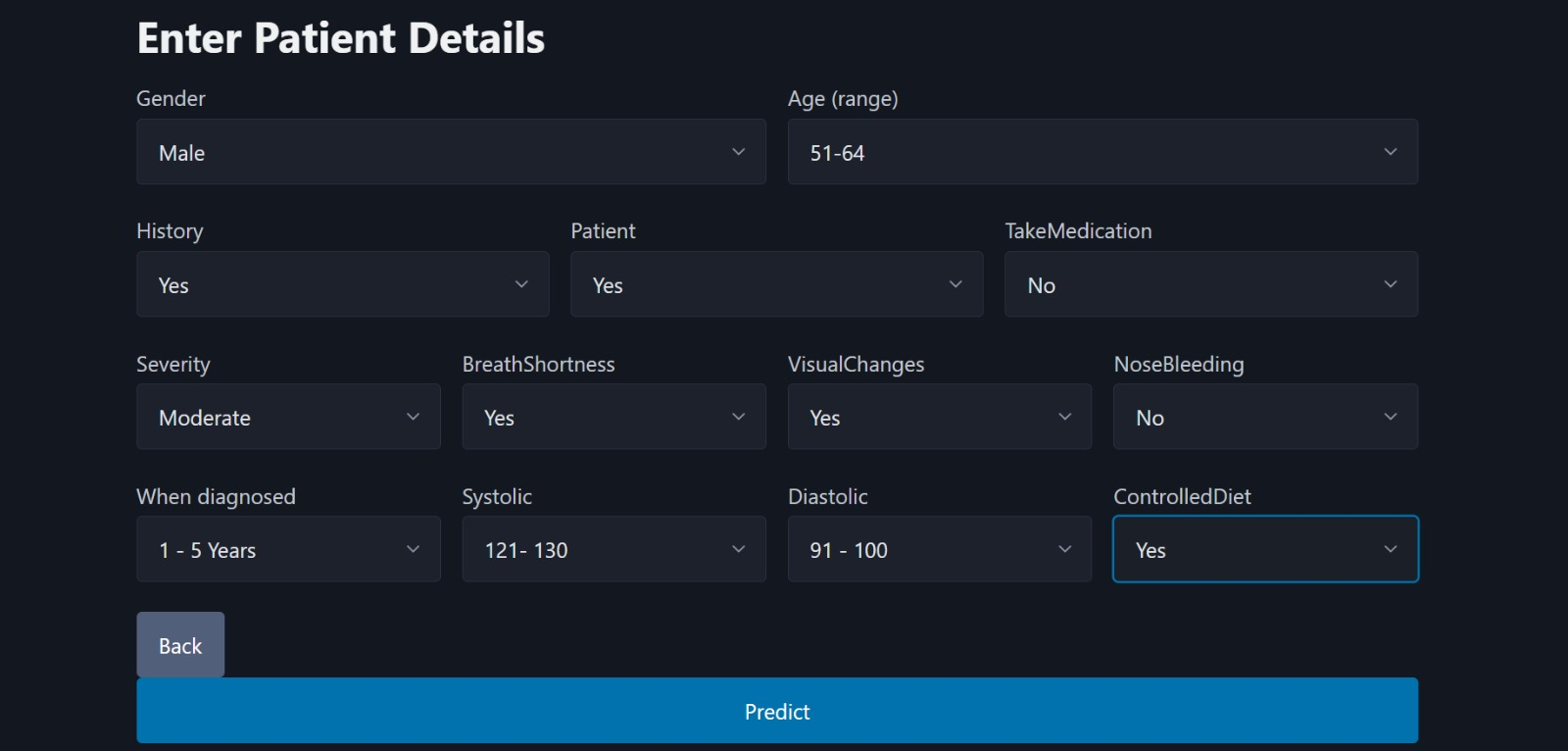
* Index Page



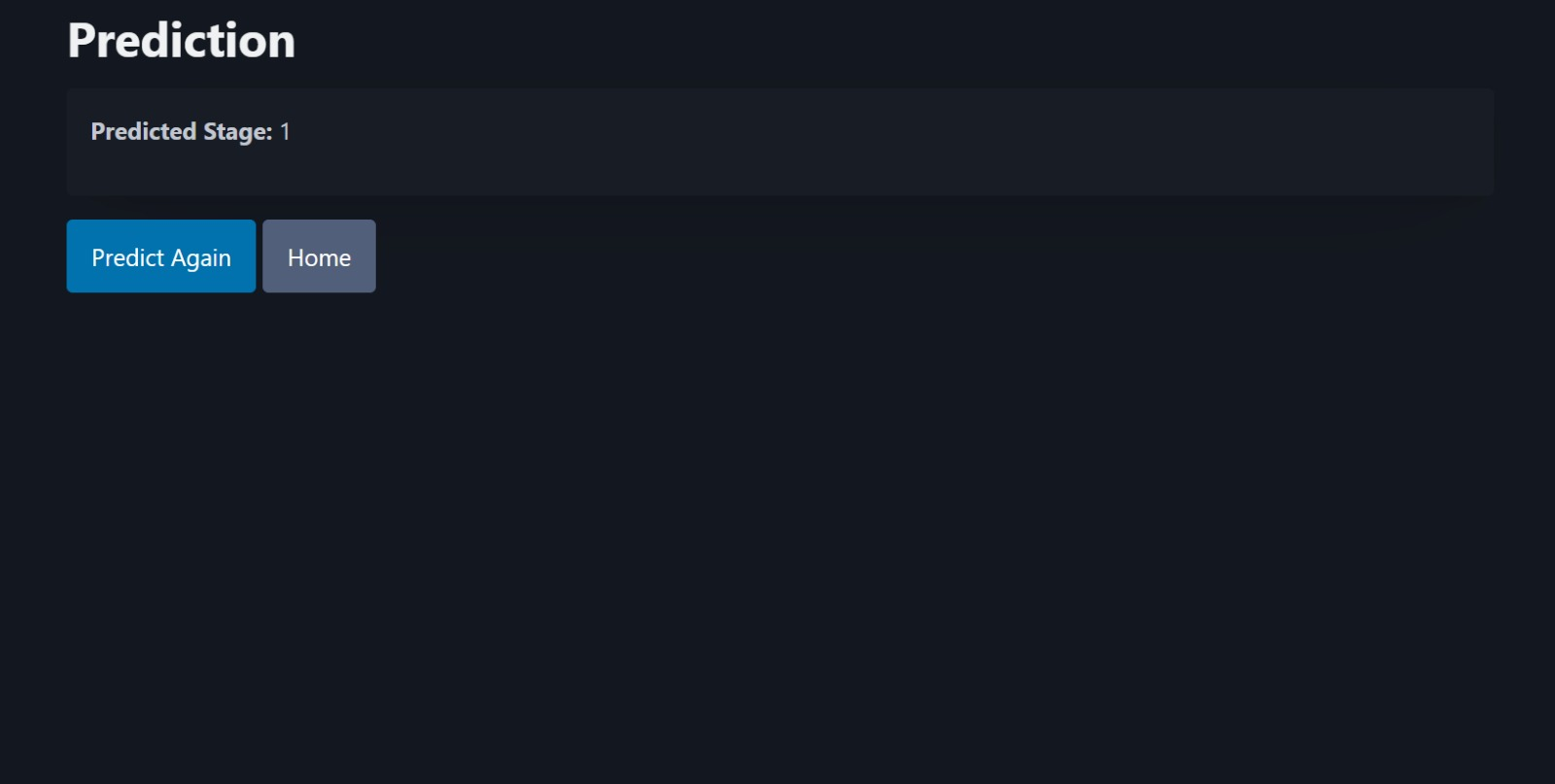


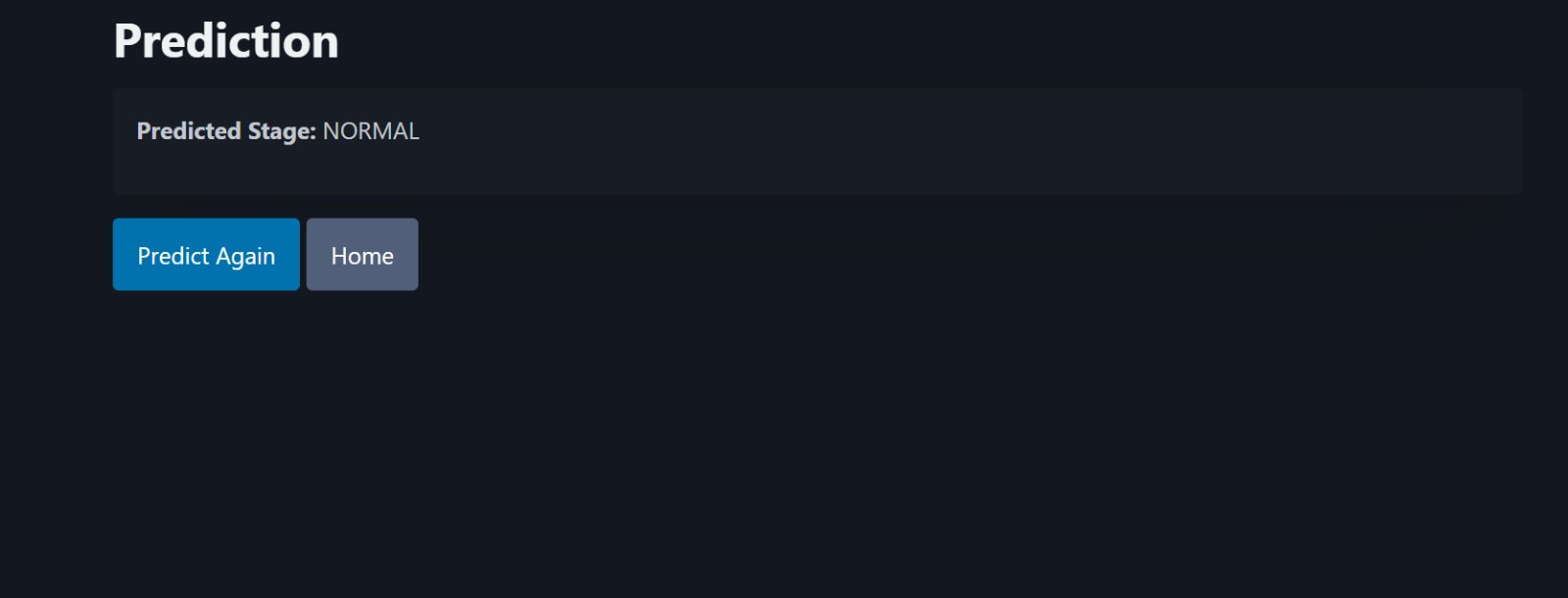


* Details Form
* Input patient health data



* Result Page
* Predicted hypertension stage





1. Future Enhancements

* Integration with a live health monitoring device API.
* User authentication for secure patient data management.
* Store prediction history in a database.
* Mobile-friendly UI.

1. Conclusion

This project demonstrates how machine learning models can be integrated into a Flask web application to provide real-time health predictions. By combining data analysis, visualization, and predictive modeling, it offers both healthcare professionals and patients an interactive tool for hypertension monitoring.