# Abstract

This report documents **Predictive Pulse — a Flask-based web** application that uses machine learning to analyze user-entered cardiovascular parameters and predict blood pressure categories and risk indicators. The application emphasizes user-friendly design, rapid inference, and explainability of model outputs.

# Table of Contents

Abstract

Milestone 1: Define Problem

Milestone 2: Data Collection & Preparation

Milestone 3: Exploratory Data Analysis

Milestone 4: Model Building

Milestone 5: Evaluation & Tuning

Milestone 6: Deployment

Milestone 7: Testing & Demonstration

Conclusion & Future Work

References

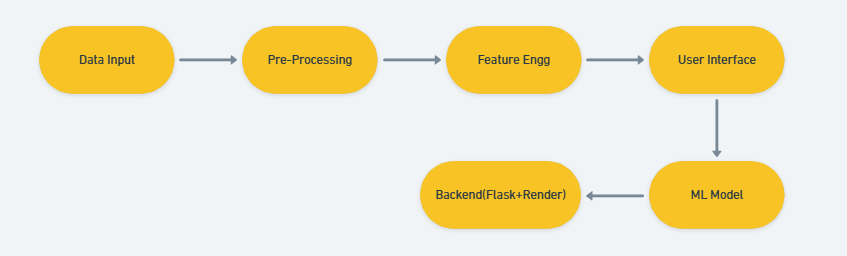
# Milestone 1: Define Problem / Problem Understanding

## Activity 1: Problem Statement & Medical Background

Hypertension (high blood pressure) is a critical risk factor for cardiovascular disease. Blood pressure is expressed as systolic/diastolic values (mmHg). Normal, elevated, and hypertension stages (Stage 1/Stage 2) classification allows early interventions. This system aims to help users self-screen and understand possible risks.

Medical context:  
- Systolic Pressure: pressure during heart contraction.  
- Diastolic Pressure: pressure during relaxation.  
- Mean Arterial Pressure (MAP) and Pulse Pressure (PP) are derived measures used in clinical evaluation.

Figure: Problem to Solution overview:



## Activity 2: Business Requirements & Stakeholders

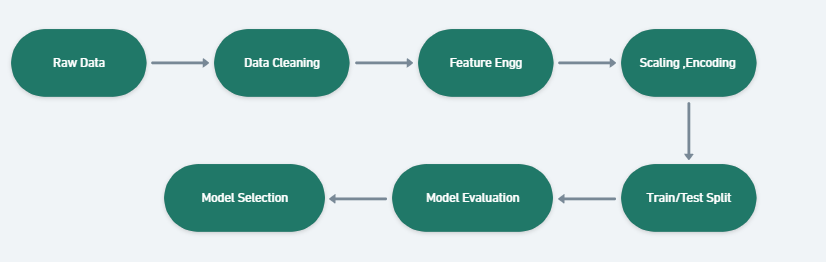
Requirements:  
- Accurate classification of blood pressure category.  
- Low-latency inference for responsive UI.  
- Secure handling and optional logging of user inputs.  
- Clear result interpretation and recommended next steps.  
  
Stakeholders include: end-users, healthcare professionals, and potential partnering telemedicine services.

# Milestone 2: Data Collection & Preparation

Data sources: The model is trained on curated datasets combining clinical blood pressure records and demographic/lifestyle features. When assembling datasets, ethical considerations (consent, anonymization) and data quality checks are essential.

Key attributes often used:  
- Age, Gender, Weight/BMI  
- Resting Heart Rate  
- Systolic & Diastolic blood pressure  
- Smoking status, physical activity, medical history

Data Preprocessing steps:



- Missing value strategies: mean/median imputation, KNN imputation for correlated features.  
- Outlier detection: IQR or Z-score for continuous variables. Apply winsorization or removal where clinically implausible.  
- Feature engineering: computing MAP, PP, categorical bucketing of age, interaction terms.  
- Scaling & encoding: StandardScaler or MinMaxScaler for numeric, one-hot/ordinal encoding for categorical.

# Milestone 3: Exploratory Data Analysis (EDA)

EDA helps understand distributions, detect data quality issues, and informs feature selection. Typical analyses include univariate histograms, boxplots for outliers, correlation heatmaps for multicollinearity, and bivariate plots.

Common findings and remediation:  
- Highly skewed measures benefit from transformation (log/Box-Cox).  
- Multicollinearity between features such as systolic & pulse pressure may require dimensionality reducion or feature selection.

# Milestone 4: Model Building

Algorithms evaluated:  
- Decision Tree: interpretable, prone to overfitting without pruning.  
- Random Forest: ensemble method reducing variance and often high-performing for tabular data.  
- Logistic Regression: baseline linear model with probabilistic outputs.  
- Support Vector Machine (SVM): effective for some decision boundaries with kernels.  
  
Selection rationale: Random Forest is often chosen for tabular health datasets due to robustness and balance between bias/variance.

Model interpretability: Feature importance from tree-based models and SHAP values can provide explanations for individual predictions.

# Milestone 5: Evaluation & Hyperparameter Tuning

Evaluation metrics for imbalanced or critical-health tasks must include precision and recall alongside accuracy. F1-score balances precision and recall. Cross-validation (k-fold) offers robust performance estimates.

Hyperparameter tuning: GridSearchCV or RandomizedSearchCV across parameters (n\_estimators, max\_depth, min\_samples\_split, etc.) with stratified folds.

# Milestone 6: Deployment

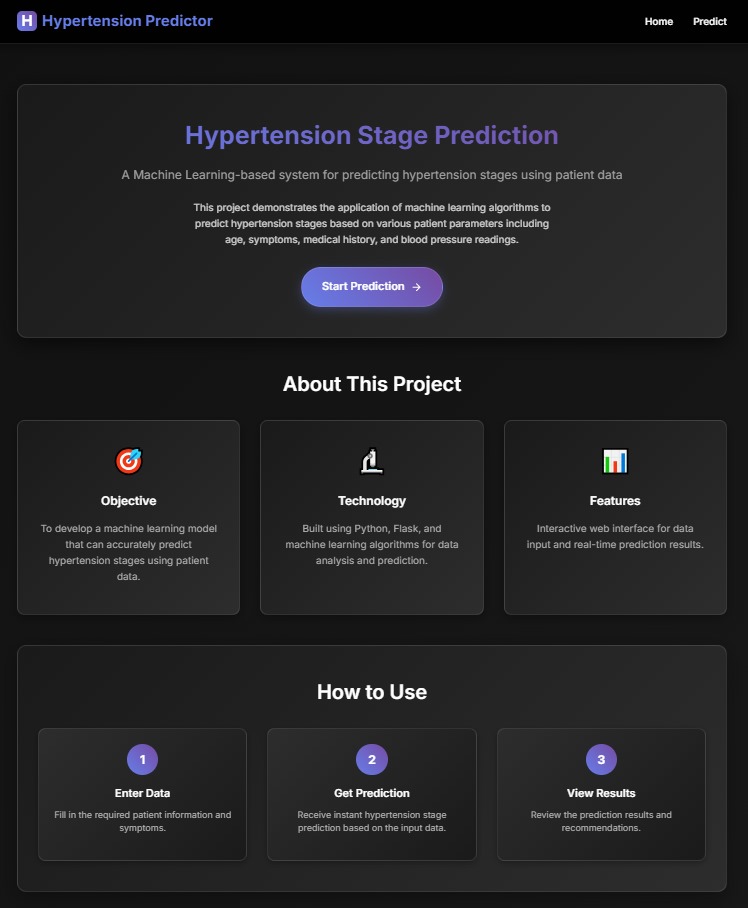
The trained model is serialized (pickle/joblib) and loaded by the Flask application on server start. REST endpoints accept form or JSON payloads, apply preprocessing pipelines, and return predictions. Deployment optimizations include caching models in memory, using gunicorn for concurrency, and containerization.

Security & Privacy: Use HTTPS, avoid logging PII, and implement user consent and data retention policies.

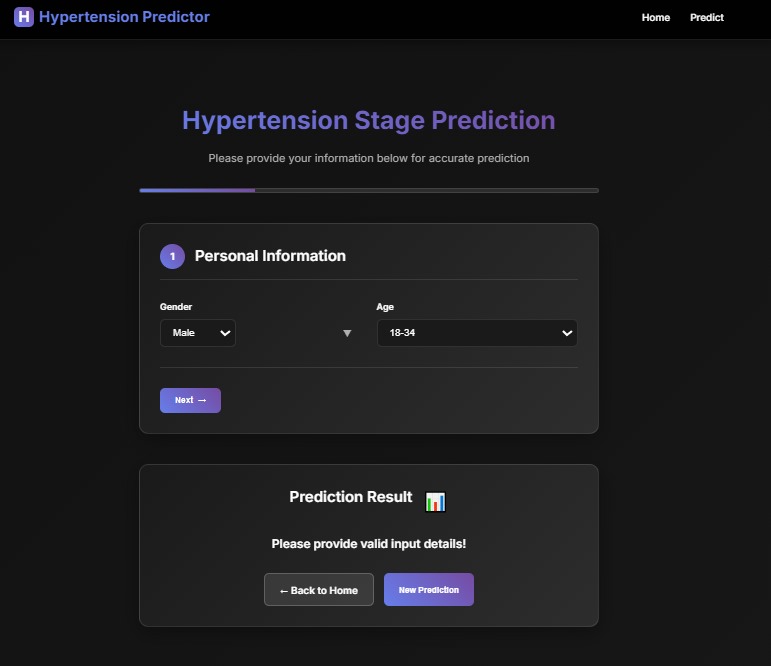
# Milestone 7: Testing, Demonstration & UX

Testing includes unit tests for preprocessing functions, model inference tests, and end-to-end UI tests. User testing assesses usability and interpretability of the results.

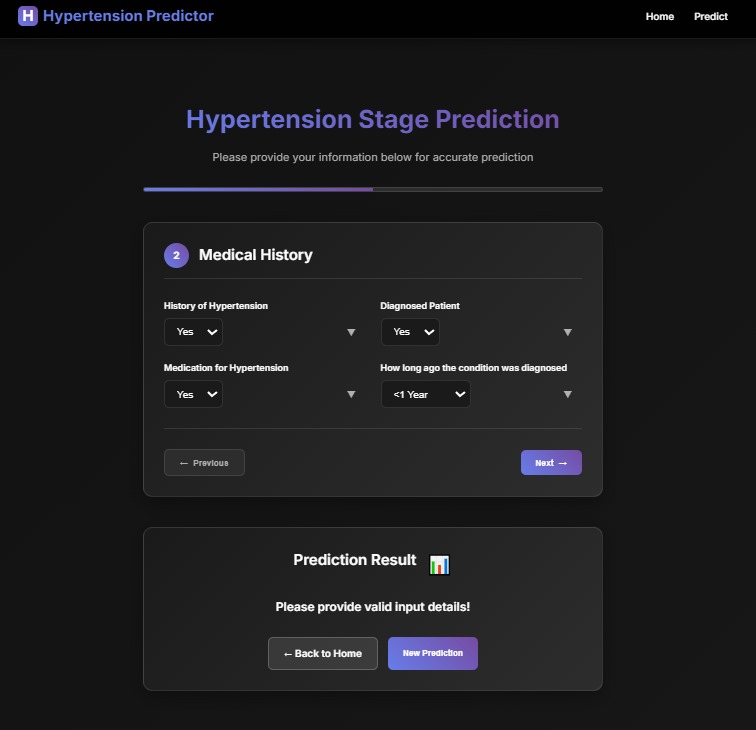
Application screenshots with captions:



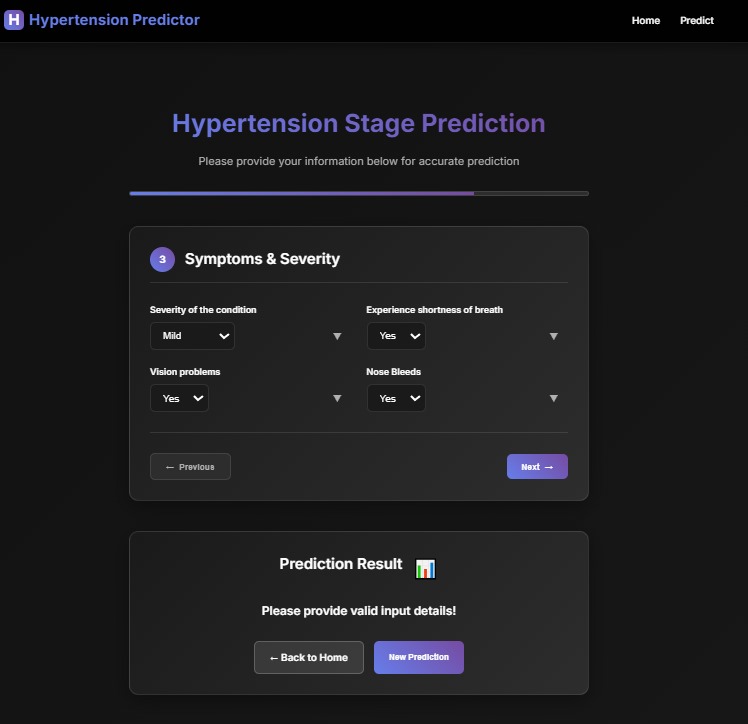
Home page & data input form.



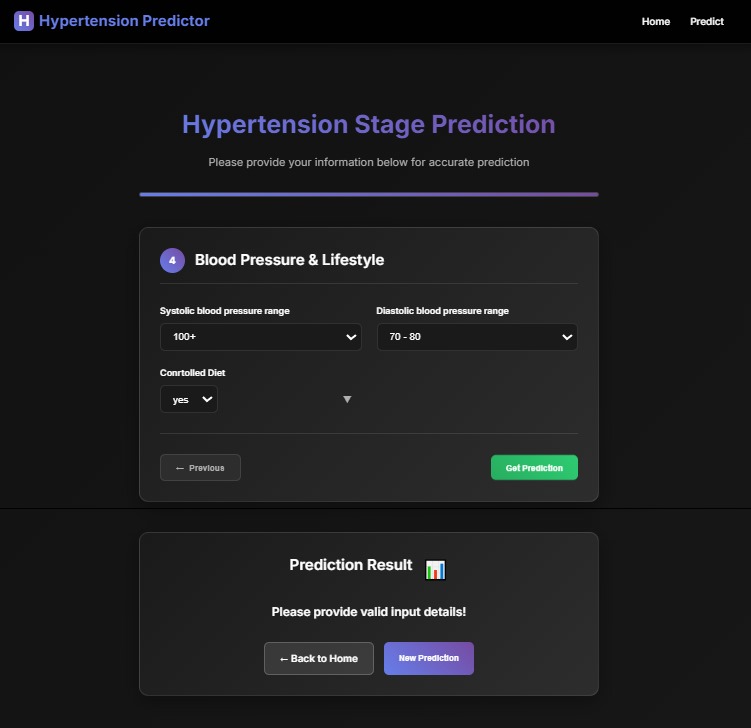
Stepwise input for demographics and vitals.

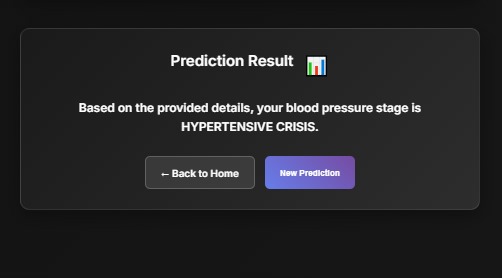


Charts and statistics displayed to users.



Model info and about page.





Final summary and recommendations.

# Conclusion & Future Scope

Predictive Pulse demonstrates an end-to-end ML solution for blood pressure analysis with a focus on early detection and user empowerment. Future work: augment training data diversity, integrate continuous monitoring from wearables, and expand model explainability using SHAP/LIME.

# References

American Heart Association — Understanding Blood Pressure

Scikit-learn documentation — Model selection & evaluation

Research articles on ML for cardiovascular risk prediction