# Heart Disease Risk Level Predictor

# Darbhanga College of Engineering


## Heart Disease Risk Level Predictor

**Presented by**

| | |
|---|---|
| **Shivansh Sagar** | 18105111016 |
| **Anjali** | 18105111031 |
| **Pragati** | 18105111034 |
| **Aman Raj** | 18105111050 |

**Guided by**

**Prof. Ajeet Kumar Gupta**

*Assistant professor, CSE Dept.*
*Darbhanga College of Engineering.*

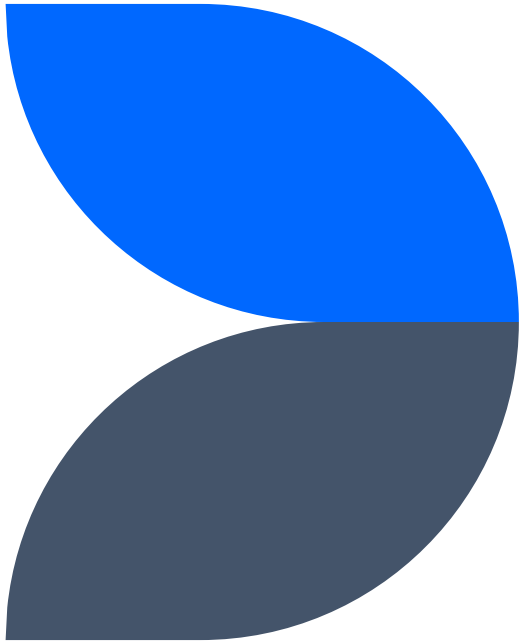
# Contents

# Introduction

Heart Disease Risk Level Predictor is a website which detect risk of heart disease build using different algorithms of Machine Learning in backend. The input to our algorithm is 8 features with number values and binary values. We use algorithms such as Linear Regression and multivariable polynomial regression to output the risk percentage which indicates the chances of having heart disease and it gives us the best accuracy of 75.8%.  and we have created a website by using html, CSS and bootstrap for taking the input of patient details and used the flask module for deploying the machine learning model and processing that data.

# SOFTWARE TOOLS & APPLICATIONS

## LIBRARIES FRAMEWORK AND COMPONENTS



# Frontend Software Technologies



GitHub

# HTML (Hyper Text Markup Language)

It is the standard markup language for creating Web pages.

HTML forms the building blocks of all websites.

It describes the structure of a Web page.

It consists of a series of elements. Its elements tell the browser how to display the content.

Some HTML tags –

 - to organize text in paragraphs.

<table></table> - to display table.

<img></img> - to add image.

# CSS (Cascading Style Sheet)

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

Set of rules defining how an html element will be “presented” in the browser.

It is designed to enable the separation of presentation and content, including layout, colors, and fonts.

CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

Things you can change with CSS – colors, spacing, font, sizes, font size, borders, background, positions.

# JavaScript

- JavaScript was developed by Netspace in 1995.
- It was originally called LiveScript.
- Classic JavaScript is a client-side language used to add interactivity your web-pages.
- **JavaScript** is used to program the behavior of web pages.
- It is used when a webpage is to be made dynamic and add special effects on pages like rollover, roll out and many types of graphics.

# Bootstrap

- Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

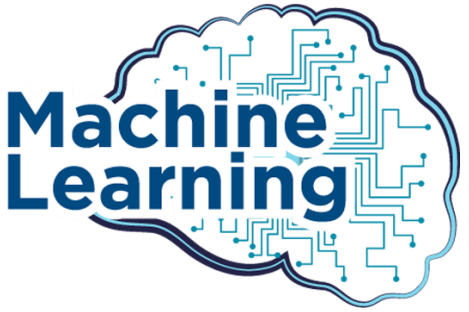
# GitHub

- GitHub, Inc. is a provider of Internet hosting for software development and version control using Git.
- It offers the distributed version control and source code management functionality of Git, plus its own features.

# Google Chrome

- Google chrome is a web browser application software for accessing the World Wide Web or a local website.
- When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then displays the page on the user's device

# Backend Software Technologies



Visual Studio Code

# Visual Studio Code

- Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS.
- Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Visual Studio Code

# Python

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.
- Python supports modules and packages, which encourages program modularity and code reuse.
- Python's simple, easy to learn syntax emphasizes readability .

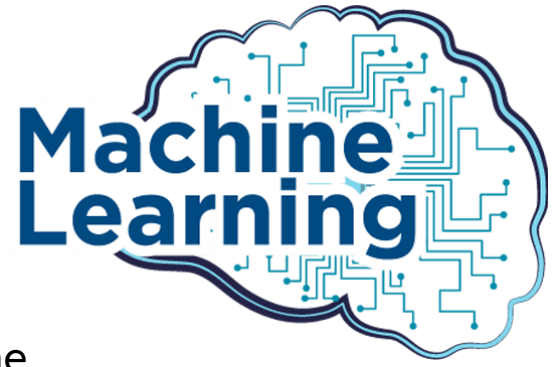
# Jupyter Notebook

- The Jupyter Notebook is an open-source web application that can be used to create and share documents that contain live code, equations, visualizations, and text.
- Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning.
- A modular design invites extensions to expand and enrich functionality.

# Machine Learning

- Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

  We used two algorithm in this project, and they are:




- **Linear Regression:** Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent (x) and dependent (y) variables.
- **Multivariable Polynomial Regression:** Multivariate Multiple Regression is the method of modeling multiple responses, or dependent variables, with a single set of predictor variables.

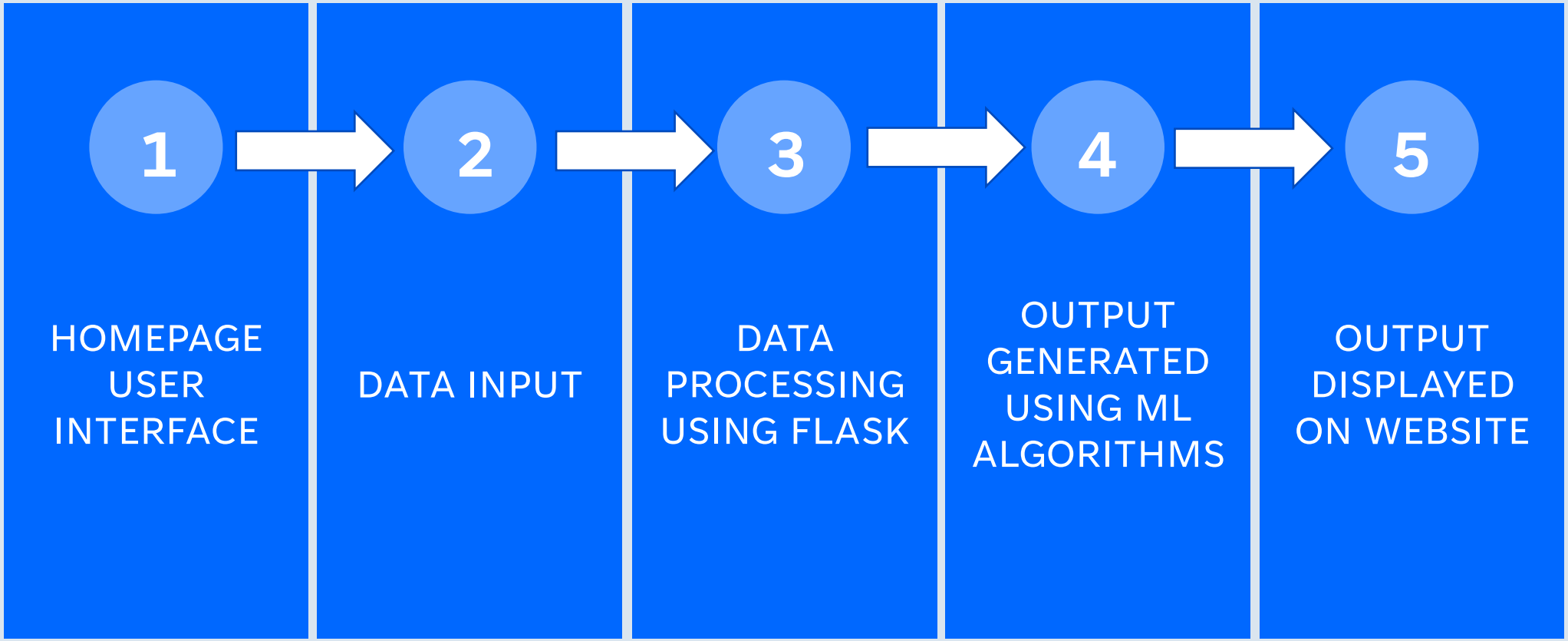
# Flask

- Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier.
- Flask uses the Jinja template engine to dynamically build HTML pages using familiar Python concepts such as variables, loops, lists, and so on. We have used these templates as part of this project.

# Libraries & Module for Algorithm Development Using Python

- asttokens==2.0.5
- backcall==0.2.0
- click==8.0.4
- colorama==0.4.4
- debugpy==1.5.1
- decorator==5.1.1
- entrypoints==0.4
- executing==0.8.3
- Flask==2.0.3
- nest-asyncio==1.5.4
- ipykernel==6.9.2
- ipython==8.1.1
- itsdangerous==2.1.1
- jedi==0.18.1
- Jinja2==3.0.3
- joblib==1.1.0
- jupyter-client==7.1.2
- jupyter-core==4.9.2
- MarkupSafe==2.1.1
- numpy==1.22.3
- pandas==1.4.1
- six==1.16.0
- sklearn==0.0
- stack-data==0.2.0
- threadpoolctl==3.1.0
- tornado==6.1
- traitlets==5.1.1
- wcwidth==0.2.5
- Werkzeug==2.0.3
- parso==0.8.3
- pickleshare==0.7.5
- Pygments==2.11.2
- python-dateutil==2.8.2
- pytz==2022.1
- pywin32==303
- pyzmq==22.3.0
- scikit-learn==1.0.2
- scipy==1.8.0
- pure-eval==0.2.2
- prompt-toolkit==3.0.28
- psutil==5.9.0
- matplotlib-inline==0.1.3

# STEPS FOR RISK PREDICTION



# Frontend Working and Building

We created a website by using HTML, CSS and Bootstrap for taking the input from the user and displaying the calculated result.

The website has several pages:-

- Home page:

  This is the first page of the website which contains the navigation bar and footer along with the (click here) button which will navigate the user to the patient detail page which contains the form.

- Patient detail page:

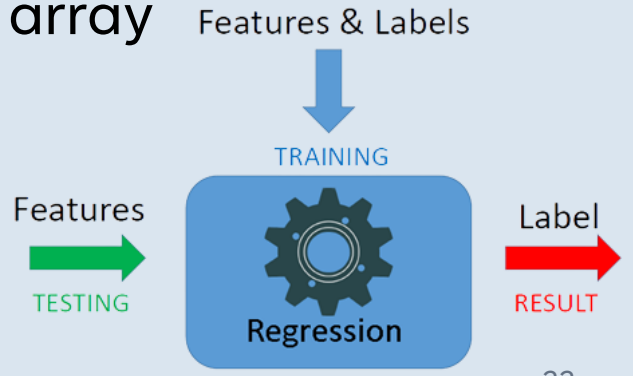
  This page contains the form which is required to be filled by the user to calculate the heart risk. It contains all the features (gender, age, tc, hdl, sbp, smoke, blood pressure medication, diab) which are required by the machine learning model to predict the result.

- Patient Result page:

  This page will display the calculated result along with some reference data which can help the user to compare his/her data with the given normal range.

# Backend Working and Building

- In the backend we have used flask(a framework of python) for deploying the machine learning model and processing that data and used two algorithms-linear regression and multivariable polynomial regression.
- The machine learning model was trained using the available data set using the linear regression and multivariable polynomial regression one by one.
- Many libraries of python was used/imported for doing this like matplotlib for data visulaization , numpy for performing array operations, pandas, sklearn .

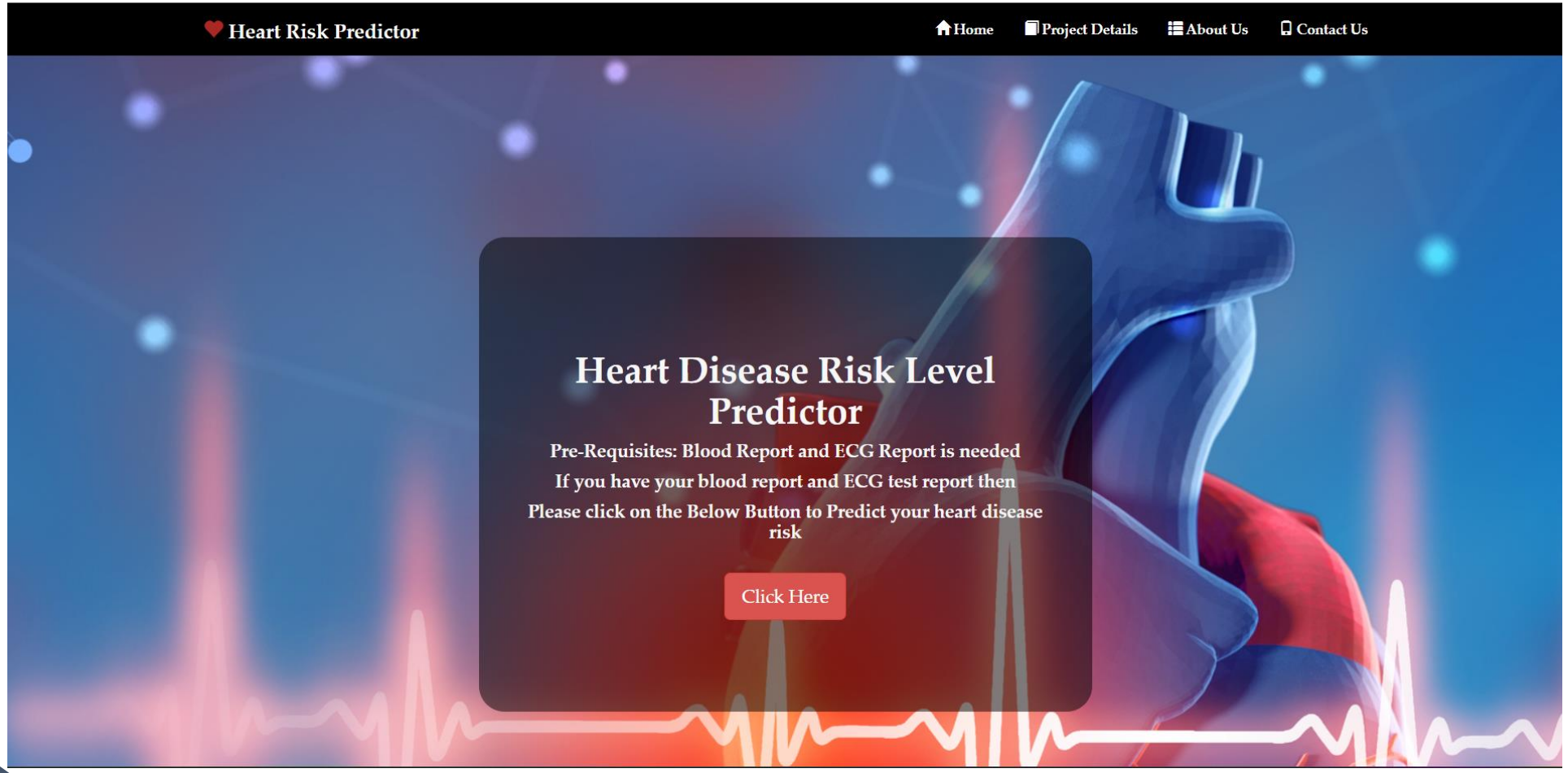


# DATASET AND FEATURES

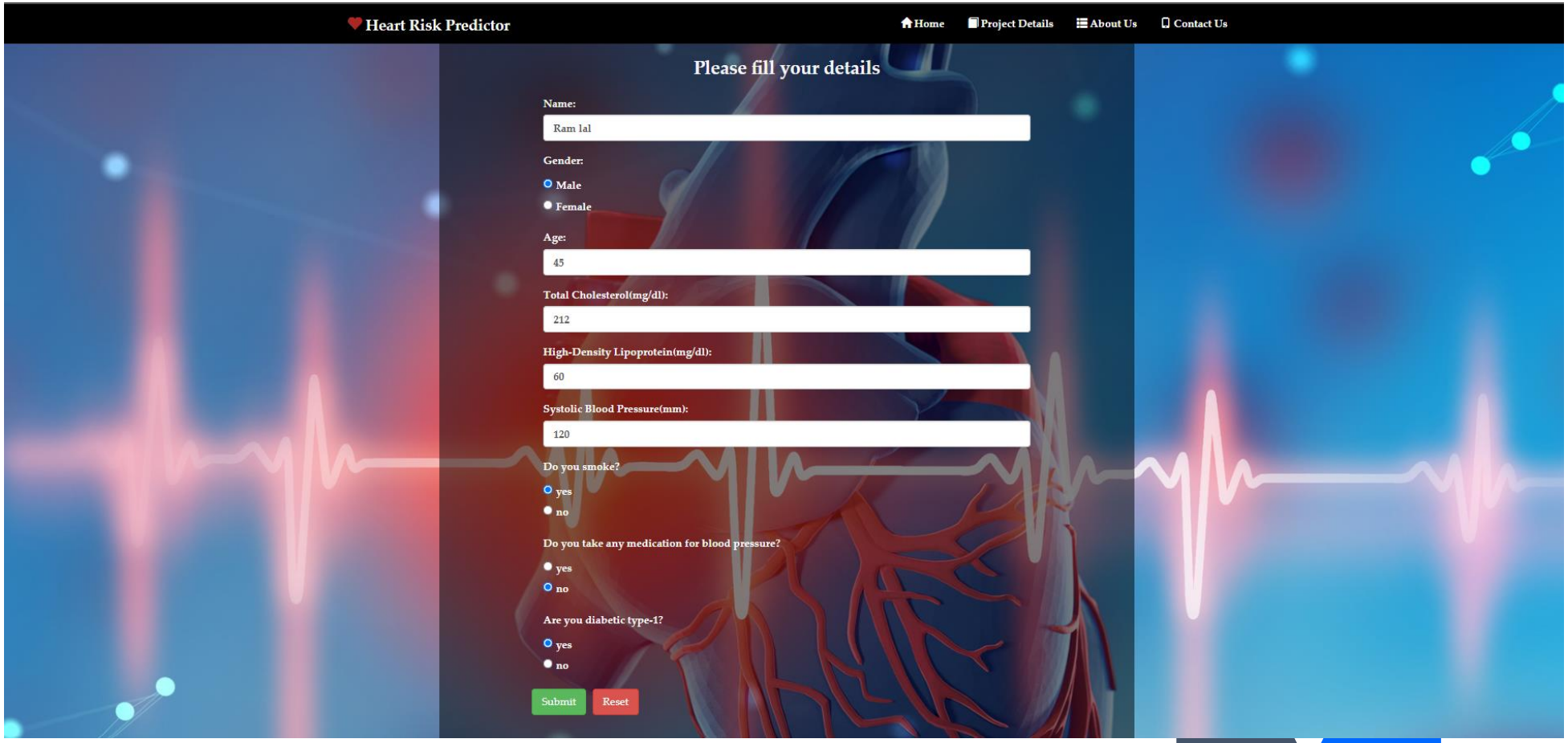
The data set for this model was taken from Kaggle (data repository) and it has 6644 instances.

- gender: gender (1=male; 2=female)
- age: age (in years)
- tc: Total cholesterol (in mg/dL)
- hdl: High-density Lipoprotein (in mg/dL)
- sbp: Systolic Blood Pressure (in mm)
- smoke: smoke (1=yes; 0=no)
- blood pressure medication: Blood Pressure Medication (1=no; 2=yes)
- diab: diabetic type 1(1=yes; 0=no)

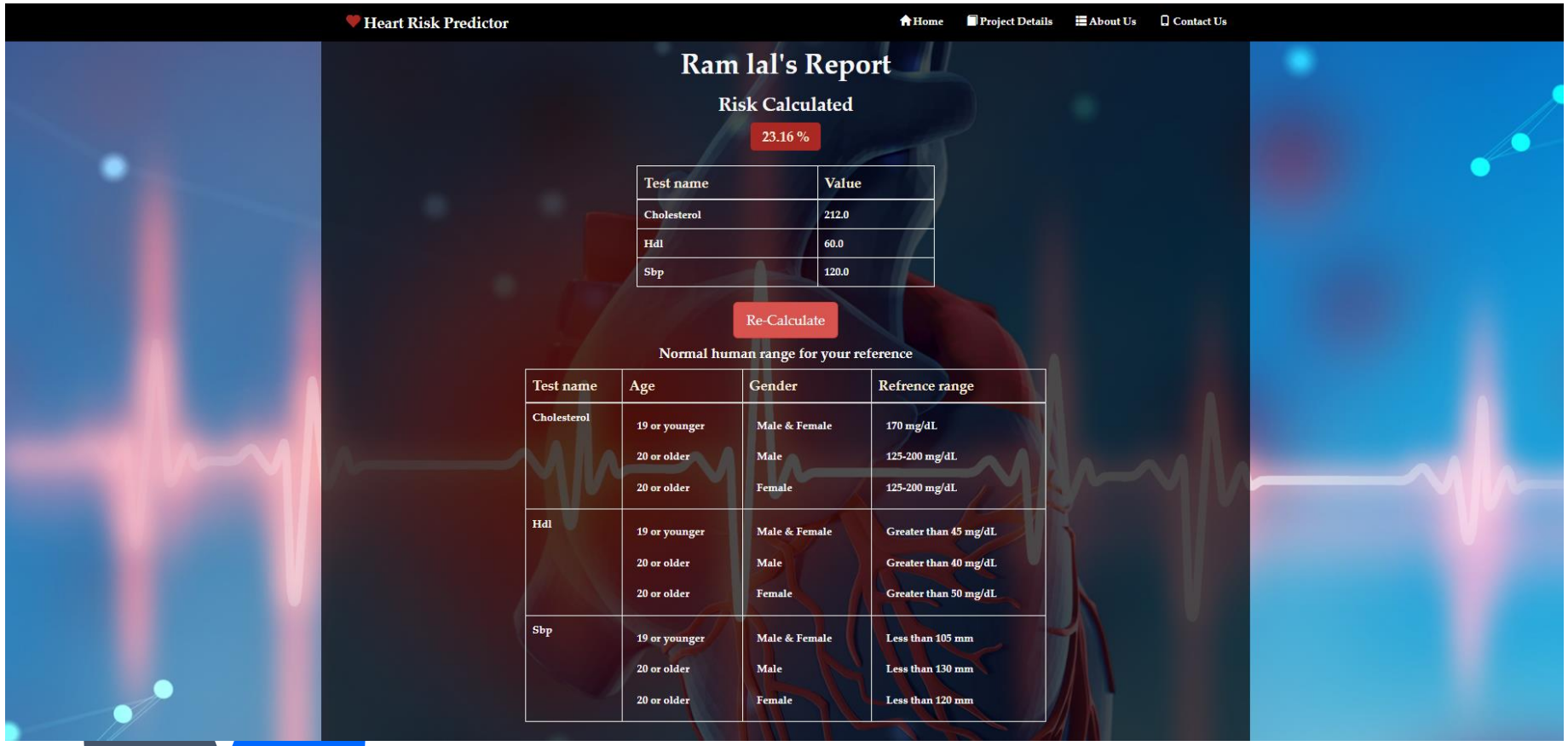
# Website Homepage User Interface



# Website Patient Details UI



# Website Patient Result UI



# References

[1] Soni, Jyoti, et al. "Predictive data mining for medical diagnosis: An overview of heart disease prediction." International Journal of Computer Applications 17.8 (2011): 43-48.

[2] Baccouche, Asma, et al. "Ensemble Deep Learning Models for Heart Disease Classification: A Case Study from Mexico." Information 11.4 (2020): 207.

[3] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[4] https://www.kaggle.com/ronitf/heart-disease-uci

# Conclusion

In this project we successfully deployed a website which can be used to predict heart disease risk level by taking patient detail as input.

We used some libraries provided by Python and html, CSS and bootstrap to implement this project. After the experiments, the algorithm of Multivariable Polynomial Regression gives us the best test accuracy, which is 75.8%. The reason why it outperforms others is that it is not limited to the property of the dataset. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

Though we get a good result of 75.8% accuracy, that is not enough because it cannot guarantee that no wrong diagnosis happens. To improve accuracy, we hope to require more dataset.

# Future Scope

- The project presently gives the result according to the pre-trained model .
- In future the project can be made to update its model and increase the dataset size in order to gain more precision .
- And more ways could be found where we can integrate heart-disease-trained ML and DL models with certain multimedia for the ease of patients and doctors for a better and reliable prediction of the cause.

# Thank you