## APPENDIX 1

**An Expert System for Predicting Heart Diseases**

PROJECT REPORT

***By***

*Divya. K (11904600)*

*Akash Sirohi (11903253) Aman Bhaskar (11903841)*

*Kanksha (11910123)*



**Department of Intelligent Systems**

**School of Computer Science Engineering**

**Lovely Professional University, Jalandhar.**

**April - 2020.**

## APPENDIX 2

**Student Declaration**

This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources has been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

Divya. K

K19BBA02

Akash Sirohi

K19BBA06

Aman Bhaskar

K19BBA13

Kanksha

K19BBA11

Place Date

**APPENDIX 3**

**TABLE OF CONTENTS**

**TITLE PAGE NO.**

|  |  |
| --- | --- |
| **1. Background and objectives of project assigned................** | **1** |
| **1.1 Abstract**....................................................................... | **1** |
| **1.2 Introduction................................................................** | **1** |
| **1.3 Objective......................................................................** | **1** |
| **1.4 Scope &Limitations....................................................** | **1** |
| **2. Description of the Project...............................................** | **1** |
| **2.1 Methodology................................................................** | **1** |
| **2.2 Approach.....................................................................** | **3** |
| **3. Description of Work Division..........................................** | **4** |
| **4. Implementation…………………………………………** | **4** |
| **4.1 Dataset Description…………………………………** | **4** |
| **4.2 Code………………………………………………….** | **6** |
| **4.3 GUI Design…………………………………………..** | **13** |
| **5. Software and Hardware Requirements………………..** | **14** |
| **6. Result…………………………………………………….** | **15** |

APPENDIX 4

BONAFIDE CERTIFICATE

Certified that this project report “An Expert System for Predicting Heart Diseases” is the

bonafide work of “Divya. K, Akash Sirohi, Aman Bhaskar and Kanksha” who carried out

the project work under my supervision.

### <<Signature of the Supervisor>>

<<Name of supervisor>>

<<Academic Designation>>

<<ID of Supervisor>>

<<Department of Supervisor>>

1. **Background and Objective:**
   1. **Abstract:**

In the heat of the moment, through the knowledge it is necessary to address certain concerns, the amount of the health information which is being generated by the fast-growing technologies is too quickly, useful data is extracted and relevant outcomes are produced for the variety of problems raised by the scientist. Predicting the disease at early stage is essential for the consultants in order to save someone’s life. Similarly, cardio disorder is one of the heart concerns of the consultants. For providing solution to respective issue, we proposed an expert system to predict the heart diseases using fuzzy model in MATLAB.

* 1. **Introduction:**

In present time utilization of digital technology is essential in every field & medical diagnosis field is not an exception. The diagnosis based upon the practice, expertise & better utilization techniques with complex and efficient conceptual reasoning capability. We know digitalized fields have high complexity and randomness and hence there is a must need of developing expert systems based on fuzzy logic, artificial neural networks, and machine learning.

* 1. **Objective**

As one of the most important cause of death worldwide is heart attack. Which is one of the results of heart diseases isn’t it? Hence diagnosis of heart disease at earlier stages is important. But as there is a lot of ambiguity and risk involved it is difficult for even human experts. In case of an uncertain heart attack the detection of heart attack at initial stage is highly essential to save the life of patient & prevent heart damage. Our project has the object to perform such operation and will work on the place of human expert in those critical situations.

* 1. **Scope and Limitation**

Research should also be done on other similar systems or other datasets to test its quality to produce higher accuracy result. In this project one can generate more practical rules from real time data in order to gain higher accuracy and less error.

1. **Description of Project:**

**2.1 Methodology:**

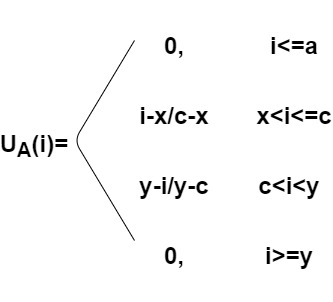
This section includes fuzzy membership functions, which have been used in the development of fuzzy expert system modeling (ranges for all input attribute) & Anfis rule base.

1. Fuzzy Membership Function

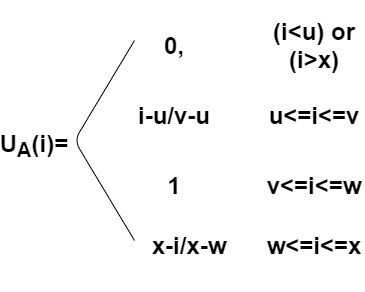
We used two membership functions for development in MATLAB

Triangular function: It is refer by a lower limit x, an upper limit y & value c, where x<c<y. It is denoted as “trimf” in MATLAB” fuzzy api.

(1)



Trapazodial function: In this function ‘u’ can be used as a lower limit and ‘x’ as upper limit with sub-lower limit ‘v’ and sub-upper limit ‘w’, where u<v<w<x. It is denoted as “trangf” in MATLAB.



1. Fuzzy Expert System Design

The most emphasizing application of fuzzy expert system is a randomness problem. Initial step in designing is determination of input and output variables. The system consists of 9 input and 1 output field. The output field computes whether the patient has heart disease or not.

1. Anfis Rule Base.

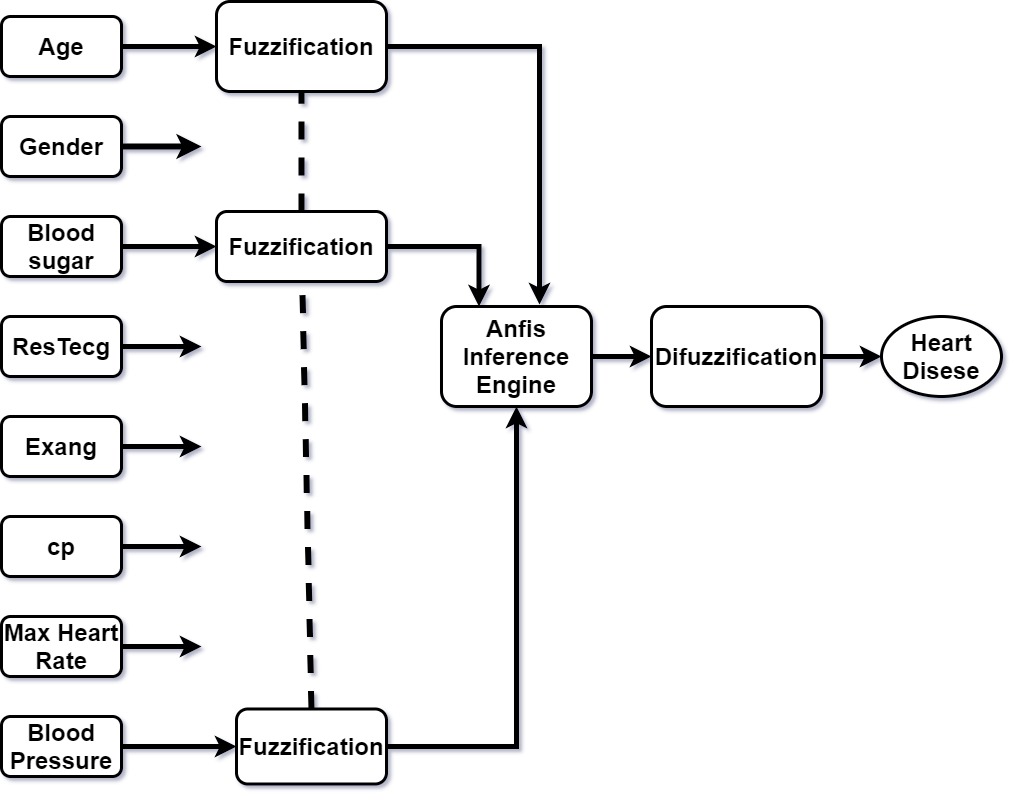
Fuzzy rule base is one of the most vital phases of fuzzy inference system and the accuracy of diagnosis depends mainly upon the rule base. It consist of single attribute or combination of attribute and having AND/OR logical operations. In this part we build 12288 rules. The whole rule base utilizes entire attributes with essential output fuzzy set.

(2)

**2.2 Approach**

To predict the heart disease, we proposed a novel dynamic approach which takes input 9 crisp input

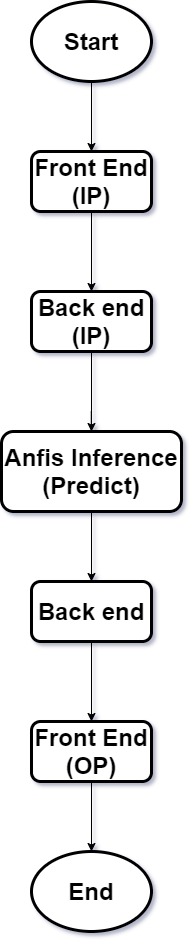
sets apply fuzzification on them. The output of fuzzification is the used for input to Anfis (Artificial neuro-fuzzy inference system) for building rules. These rules are the used to train the model and find the output function. The output value undergoes defuzzification. This leads to crisp 0 or 1 value for our model to predict heart disease.



1. **Work Flow diagram**

The work flow structure of the project is as follows:-

(3)

****

1. **Description of Work Division:**

Coding and linking of the system is handled by Akash Sirohi, Fis file generation is done by Aman, Report prepared by Divya. K, and GUI design done by Kanksha.

1. **Implementation:**

**4.1 Dataset**

The dataset is collected from kaggle website referred to as “Heart Disease UCI “. Dataset contains 14 attributes in which 9 are utilized. The aim of the dataset is predict heart disease in the patients.

**Content**

**Attribute Information:**

1. **Age:** It states the age of the patient under the test or observation. The range of the age is from (25-99) years. Age is an important feature in predicting the heart disease as with raise of age the chances of person suffering from any heart disease also increases.
2. **Gender:** It states the gender of the patient under observation. This is a categorical attribute having two values i.e. either “male” or “female” (0, 1). It is seen that the chances of heart disease in patients depends upon their gender as well.
3. **Chest Pain cp:** It states does the patient having a symptom of chest pain. It has 4 values which state the frequency of chest pain experience i.e. (0,1,2,and 3).

**0: Typical Angina 1: Atypical Angina 2: Non-Angina pain 3: Asymptomatic**

(4)

1. **Blood Pressure:** It states the measure of resting/normal blood pressure of patient body. It is an important indicator of heart disease. Its range is from 94 mm Hg to 200 mm Hg.
2. **Blood Sugar:** It states the fasting Blood sugar of the patient under observation. It should be greater than 120 mg/dl. Considering the point the attribute have binary value i.e. 1=yes; 0 =no.
3. **Maximum heart rate achieved:** It states the value of maximum heart rate achieved in the measuring the heart rate of patient under normal circumstances.
4. **Cholesterol**: It states the serum cholesterol level in patient body. It is important for predicting heart disease in patient as it said that higher the cholesterol level higher the chances of blood pressure diseases.
5. **Resting electrocardiographic results:** It is the measure of test of performed on patient in normal resting position known as electrocardiographic test. It is one of the indicator of heart disease in people having positive result than 1 else 0.
6. **Exercise induced angina:** It is the strain induced in muscle when there is an intense work out or exercise performed by the patient in observation. If yes 1 else 0.

All above mentioned attributes are used as a fuzzy input sets for our project and we then build a prediction model on the basis of these features by assigning various member functions. The output of the model is binary value showing patient is suffering from heart disease or not. If yes the how much percent the spread of disease and for no vice versa.

**Files Used In MATLAB:**

The following types of files are used in our prediction project. These files are liked to form the basic structure of our project. These files can be categorized as front end and back end files.

**Front End files:** These are the files which build the user interface of the project and these are required for display or results:

* **Input & output (.m):** It is the matlab file which contains the function definition which connects the user GUI (Graphic User Interface) file to the back end files. The elements of GUI file are using callto these functions to operatesdifferent parts of project.
* **Input & output (.fig):** Its basically the drop and place feature of matlabwhich automatically generate these files when guide is called in command window.

**Back End Files:** These are the files which build the prediction model of the project and these required for working of project:

* **Project (.fis):** It is the fuzzy input file which contains the member function types i.e. triangular, trapezium etc. which connects the user crisp input to the AnfisSugeno inference engine. The elements of GUI file are usedto take crisp value for predicting output.

**Function Used In Matlab:**

* **fuzzy :** It is used to open the fuzzy interface.
* **guide :** It is used to create the gui.
* **global :** It is used to create the global variable.

(5)

* **str2num(“string”)** : It is used to convert the string value to numeric.
* **get() :** It is used to retrieve the value from gui.
* **set() :** It is used to set the values to the gui.
* **eventdata:** It is used to handle the event of selecting the radio button.
* **readfis(“a.fis”) :** It is used to read the “fis” file i.e. fuzzy file.
* **evalfis(input,fis):** It is used to evaluate the output of the fis file by passing the two argument one is input values and other is fis file.
* **setappdata(0,’v’,b):**It is used to transfer the values from one gui to other. Basically it stored the value in a particular session as a session id. 0 is the value remains the first argument, ‘v’ is the variable in which value is going to be stored and b is the variable that containing the value.
* **findobj():** It is used to find the object of other gui, so that a connection can be established and values can be transfer between them with the help of “setappdata()”.
* **getappdata(0,”v”):** It is used to retrieve the stored value of variable v.
* **bar(data):** It is used to draw the bar graph according to the data.
* **set(gca,”XTickLable”,{list of categorical lables of x-axis}):** to set the property of x and y axes.

**4.2 Code:**

**HomeScreen.m:**

functionvarargout = homescreen(varargin)

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @homescreen\_OpeningFcn, ...

'gui\_OutputFcn', @homescreen\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

ifnargin&&ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

ifnargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before homescreen is made visible.

functionhomescreen\_OpeningFcn(hObject, eventdata, handles, varargin)

(6)

% This function has no output args, see OutputFcn.

% Choose default command line output for homescreen

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

ha=axes('units','normalized',... %to resize the image according to axes size

'position',[0 0 1 1]);

i = imread('heart3.jpg'); % used to read image

hi=imagesc(i)

% --- Outputs from this function are returned to the command line.

functionvarargout = homescreen\_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

% --- Executes on button press in start.

functionstart\_Callback(hObject, eventdata, handles)

close()

inputscreen() % call next screen

**inputscreen.m:**

functionvarargout = inputscreen(varargin)

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @inputscreen\_OpeningFcn, ...

'gui\_OutputFcn', @inputscreen\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

ifnargin&&ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

ifnargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before inputscreen is made visible.

functioninputscreen\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

(7)

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to inputscreen (see VARARGIN)

% Choose default command line output for inputscreen

handles.output = hObject;

guidata(hObject, handles);

ha=axes('units','normalized',...

'position',[0 0 1 1]);

i = imread('heart3.jpg');

hi=imagesc(i)

% --- Outputs from this function are returned to the command line.

functionvarargout = inputscreen\_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure

varargout{1} = handles.output;

function uipanel1\_CreateFcn(hObject, eventdata, handles)

ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on selection change in age.

functionage\_Callback(hObject, eventdata, handles)

functionage\_CreateFcn(hObject, eventdata, handles)

ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

functionbp\_Callback(hObject, eventdata, handles)

functionbp\_CreateFcn(hObject, eventdata, handles)

ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

functionchol\_Callback(hObject, eventdata, handles)

functionchol\_CreateFcn(hObject, eventdata, handles)

ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

functioncp\_Callback(hObject, eventdata, handles)

functioncp\_CreateFcn(hObject, eventdata, handles)

ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

(8)

end

functionhb\_Callback(hObject, eventdata, handles)

functionhb\_CreateFcn(hObject, eventdata, handles)

ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

global gender1 %....making variable global

global exang1

global res1

global bs1

% --- Executes when selected object is changed in uipanel2.

function uipanel2\_SelectionChangeFcn(hObject, eventdata, handles) % use to retrieve the selected radio button value

global gender1

switch(get(eventdata.NewValue,'Tag'))

case'male'

gender1=1;

case'female'

gender1=0;

end

% --- Executes when selected object is changed in uipanel3.

function uipanel3\_SelectionChangeFcn(hObject, eventdata, handles)

global bs1

switch(get(eventdata.NewValue,'Tag'))

case'byes'

bs1=1;

case'bno'

bs1=0;

end

% --- Executes when selected object is changed in uipanel4.

function uipanel4\_SelectionChangeFcn(hObject, eventdata, handles)

global res1

switch(get(eventdata.NewValue,'Tag'))

case'ryes'

res1=1;

case'rno'

res1=0;

end

(9)

% --- Executes when selected object is changed in uipanel5.

function uipanel5\_SelectionChangeFcn(hObject, eventdata, handles)

global exang1

switch(get(eventdata.NewValue,'Tag'))

case'eyes'

exang1=1;

case'eno'

exang1=0;

end

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

global gender1

global exang1

global res1

global bs1

age1=str2num(get(handles.age,'string')) % retrieve the value of edit box from gui

bp1 = str2num(get(handles.bp,'string'))

chol1=str2num(get(handles.chol,'string'))

hb1=str2num(get(handles.hb,'string'))

content= get(handles.cp,'value') % retrieve the selected value from listbox

switch content

case 1

cpt1=0

case 2

cpt1=1

case 3

cpt1=2

case 4

cpt1=3

end

in=[age1,gender1,bp1,chol1,bs1,cpt1,res1,exang1,hb1]

fis=readfis('project') % read fis file

h=waitbar(0,'Please wait...','CreateCancelBtn','close'); % for adding progress bar

fori=1:2000

waitbar(i/2000)

end

close(h)

out=evalfis(in,fis) % getting output

per=out\*100

if(per<0)

per=-per;

end

in=[age1,gender1,bp1,chol1,bs1,cpt1,res1,exang1,hb1,per]

setappdata(0,'in',in)

if per>=0 && per<=40

st='No'

(10)

elseif per>40 && per<=60

st='May or May not be'

else

st='Yes'

end

setappdata(0,'per',per) % set the per value, used in next gui screen

setappdata(0,'st',st)

close()

OutputScreen() % calling the next gui screen

% --------------------------------------------------------------------

function Untitled\_1\_Callback(hObject, eventdata, ~)

% --- Executes on button press in reset.

functionreset\_Callback(hObject, eventdata, handles) % to rest all editbox

set(handles.age,'string','');

set(handles.hb,'string','');

set(handles.chol,'string','');

set(handles.bp,'string','');

% --- Executes on button press in home.

functionhome\_Callback(hObject, eventdata, handles) % to redirect to home screen

close()

homescreen()

**Outputscreen.m:**

functionvarargout = OutputScreen(varargin)

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @OutputScreen\_OpeningFcn, ...

'gui\_OutputFcn', @OutputScreen\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

ifnargin&&ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

ifnargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

(11)

% --- Executes just before OutputScreen is made visible.

functionOutputScreen\_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

ha=axes('units','normalized',...

'position',[0 0 1 1]);

i = imread('heart3.jpg');

hi=imagesc(i)

set(ha,'handlevisibility','off',...

'visible','off')

% --- Outputs from this function are returned to the command line.

functionvarargout = OutputScreen\_OutputFcn(hObject, eventdata, handles)

h= findobj('tag','uipanel1') % use to connect the other screen so that data transfer is possible

% Get default command line output from handles structure

varargout{1} = handles.output;

function edit1\_Callback(hObject, eventdata, handles)

function edit1\_CreateFcn(hObject, eventdata, handles)

ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit2\_Callback(hObject, eventdata, handles)

function edit2\_CreateFcn(hObject, eventdata, handles)

ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function axes1\_CreateFcn(hObject, eventdata, handles)

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles) % to show the result

per=getappdata(0,'per') %getting the transfer value

st=getappdata(0,'st')

set(handles.edit2,'string',per) % set the values in editbox

set(handles.edit1,'string',st)

in = getappdata(0,'in');

axes('position',[0.6,0.35,0.4,0.4])

bar(in)

set(gca,'XTickLabel',{'age','gender','bp','chol','bs','cpt','res','exang','hb','output'},'xcolor','r','ycolor','b'); % to set x-axis lable

% --- Executes on button press in back.

functionback\_Callback(hObject, eventdata, handles) % to redirect to inputscreen

(12)

close()

inputscreen()

% --- Executes on button press in home.

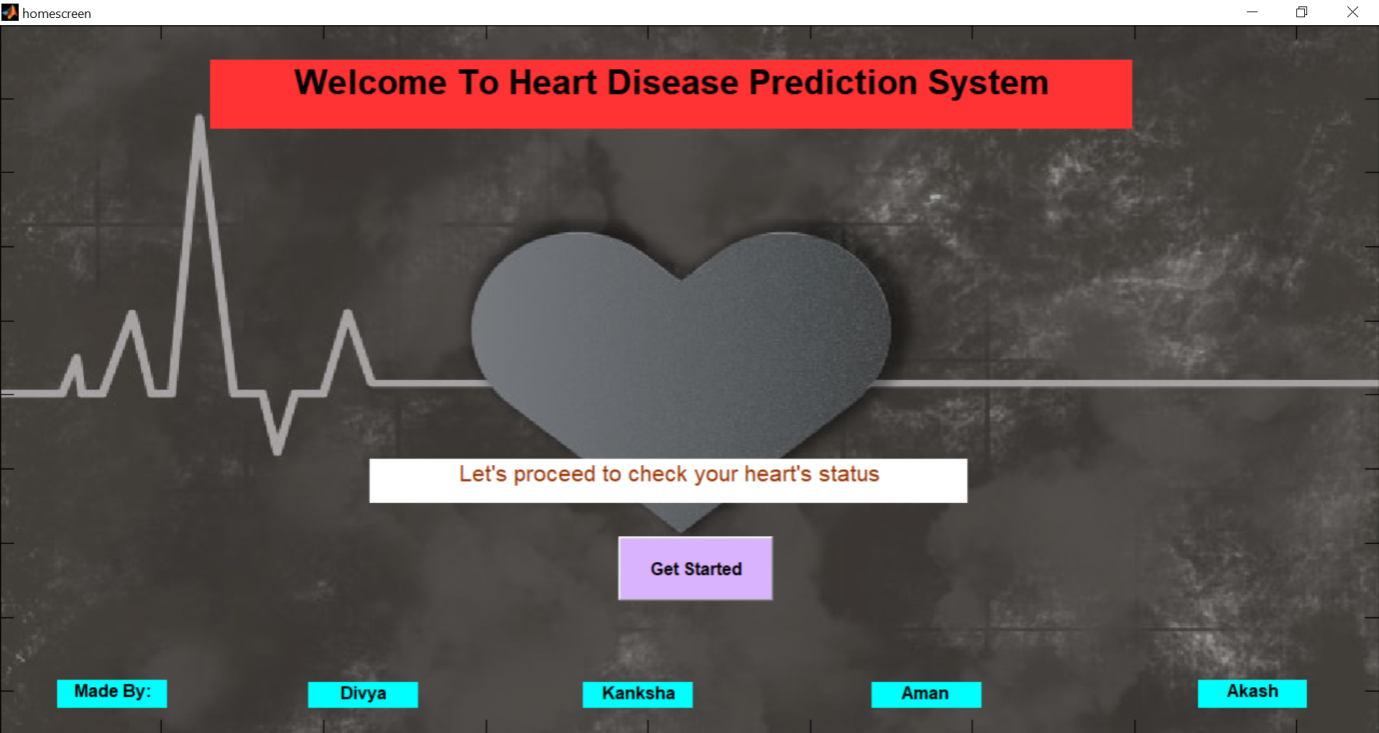
functionhome\_Callback(hObject, eventdata, handles) % to redirect homescreen

close()

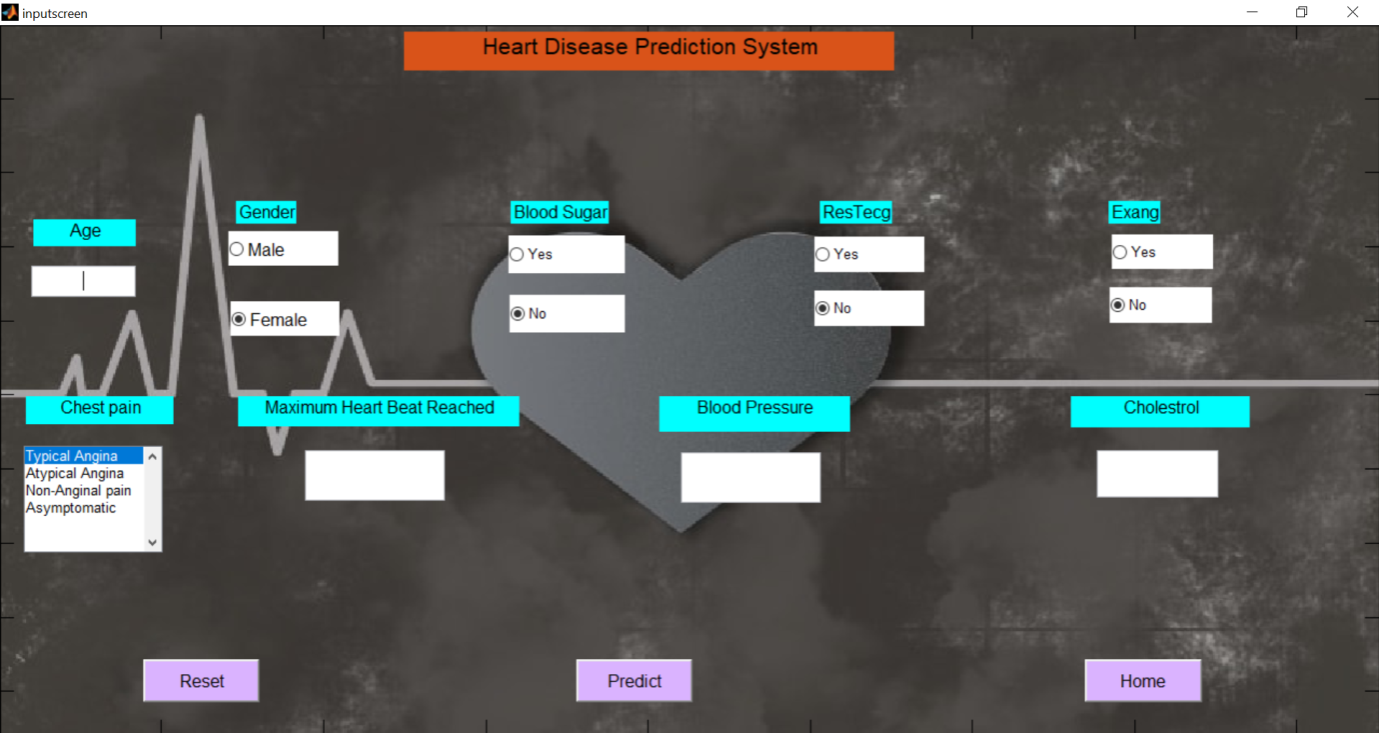
homescreen()

**4.3 GUI:**

**HomeScreen:**

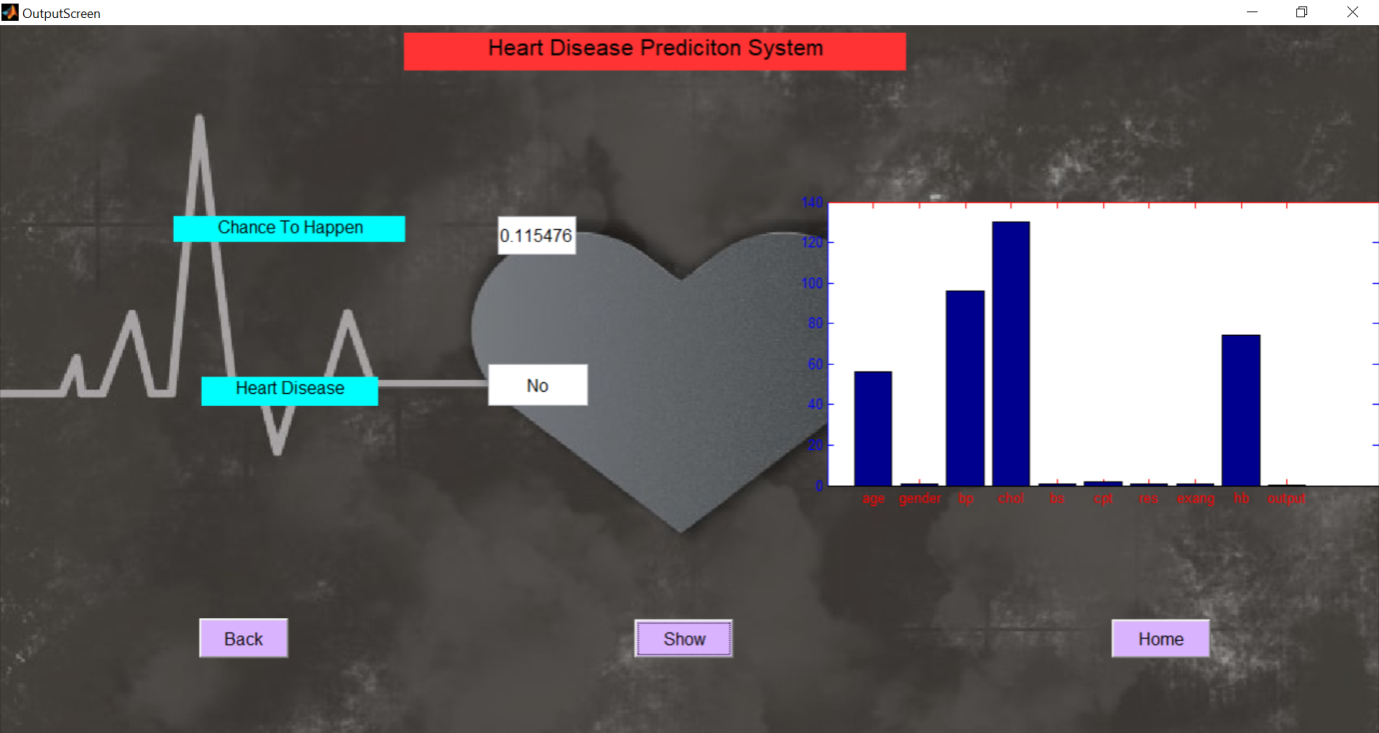
****

**INPUTSCREEN.M:**

****

(13)

**OutputScreen.m:**

****

1. **Software and Hardware Requirements:**

The software requirement specification document contains all necessary requirements for project developments.

* **Development Requirement**

**Software development**

MATLAB R2014

Operating System Windows 7,8

Anfis training environment

**Hardware Requirement**

Computer system (108 key keyboard, Monitor, CPU and mouse)

Processor - Pentium 4 and above

Memory - above 80 GB

(14)

RAM - 512 MB and above

* **User Requirement**

**Software requirement**

Operating system (System software): Windows 7,8

**Hardware requirement**

RAM, windows (version xp-128MB, vista-512MB,7 and 8-1GB)

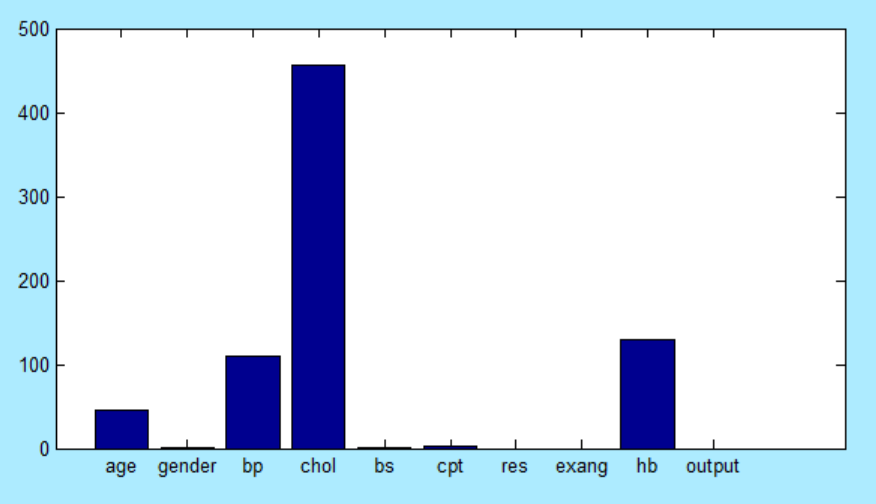
Computer system (108 key keyboard, Monitor, CPU and mouse)

Hard disk-40GB and above

Free space for application – 20MB and above

1. **Result:**

In the proposed project, various attributes are taken which are being listed in dataset description. On x-axis the attributes are mentioned and y-axis their range. Once these attribute are entered by the users, the output is predicted in terms of 0 and 1, if it predict as 0 means the person is not suffering from any heart disease and if 1 means he has heart problems but for better understandability those output values are normalized by converting into percentage. Therefore, in this way we can able to predict whether a person has heart problems or not.



(15)