IMPORT NUMPY, PANDAS ,MATPLOTLIB

In [2]:
```
# IMPORT NUMPY, PANDAS ,MATPLOTLIB
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

READ CSV FILE

In [3]:
```
# READ CSV FILE
data = pd.read_csv('heart.csv')
```

In [33]:
```
data
```

Out[33]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	Exe
0	40	M	ATA	140	289	0	Normal	172	
1	49	F	NAP	160	180	0	Normal	156	
2	37	M	ATA	130	283	0	ST	98	
3	48	F	ASY	138	214	0	Normal	108	
4	54	M	NAP	150	195	0	Normal	122	
...	...	...	...	...	...	...	...	...	
913	45	M	TA	110	264	0	Normal	132	
914	68	M	ASY	144	193	1	Normal	141	
915	57	M	ASY	130	131	0	Normal	115	
916	57	F	ATA	130	236	0	LVH	174	
917	38	M	NAP	138	175	0	Normal	173	

918 rows × 12 columns

DROP NAN VALUES

In [4]:
```
# DROP NAN VALUES
data.dropna()
```

Out[4]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	Exe
0	40	M	ATA	140	289	0	Normal	172	
1	49	F	NAP	160	180	0	Normal	156	
2	37	M	ATA	130	283	0	ST	98	
3	48	F	ASY	138	214	0	Normal	108	
4	54	M	NAP	150	195	0	Normal	122	
...	...	...	...	...	...	...	...	...	
913	45	M	TA	110	264	0	Normal	132	
914	68	M	ASY	144	193	1	Normal	141	
915	57	M	ASY	130	131	0	Normal	115	
916	57	F	ATA	130	236	0	LVH	174	
917	38	M	NAP	138	175	0	Normal	173	

918 rows × 12 columns

DROP DUPLICATES

In [5]:
```
# DROP DUPLICATES
data.drop_duplicates()
```

Out[5]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	Exe
0	40	M	ATA	140	289	0	Normal	172	
1	49	F	NAP	160	180	0	Normal	156	
2	37	M	ATA	130	283	0	ST	98	
3	48	F	ASY	138	214	0	Normal	108	
4	54	M	NAP	150	195	0	Normal	122	
...	...	...	...	...	...	...	...	...	
913	45	M	TA	110	264	0	Normal	132	
914	68	M	ASY	144	193	1	Normal	141	
915	57	M	ASY	130	131	0	Normal	115	
916	57	F	ATA	130	236	0	LVH	174	
917	38	M	NAP	138	175	0	Normal	173	

918 rows × 12 columns

OBSERVE INFO OF THE DATA

In [6]:
```
# INFO OF THE DATA
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Age             918 non-null    int64  
 1   Sex             918 non-null    object 
 2   ChestPainType   918 non-null    object 
 3   RestingBP       918 non-null    int64  
 4   Cholesterol     918 non-null    int64  
 5   FastingBS       918 non-null    int64  
 6   RestingECG      918 non-null    object 
 7   MaxHR           918 non-null    int64  
 8   ExerciseAngina  918 non-null    object 
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object 
 11  HeartDisease    918 non-null    int64  
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

DESCRIBE DATA

In [7]:
```
#  DESCRIBE DATA
data.describe()
```

Out[7]:

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

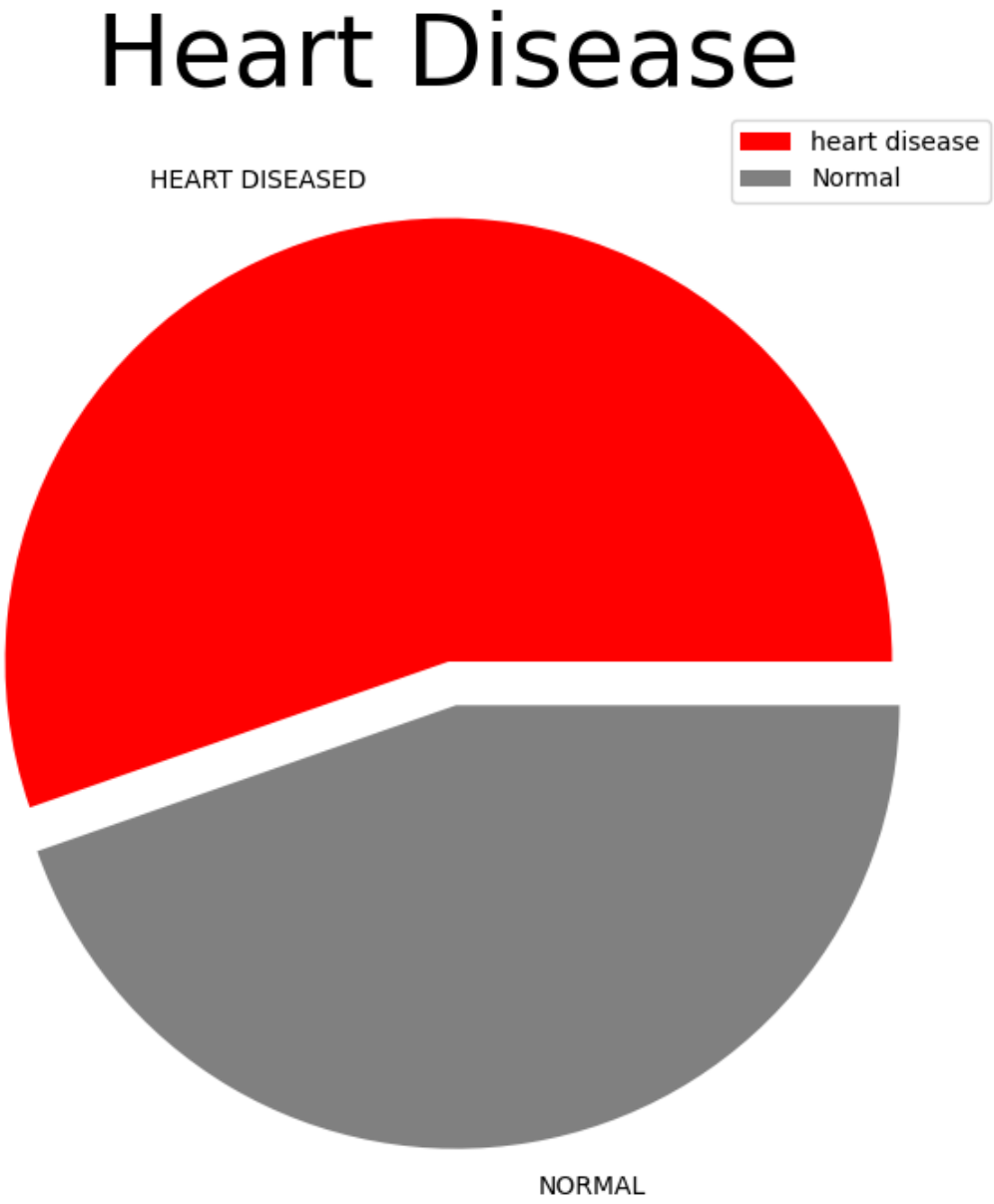
In [8]:
```
data.isna().sum()
```

Out[8]:
```
Age              0
Sex              0
ChestPainType    0
RestingBP        0
Cholesterol      0
FastingBS        0
RestingECG       0
MaxHR            0
ExerciseAngina   0
Oldpeak          0
ST_Slope         0
HeartDisease     0
dtype: int64
```

HEART DISEASED AND NORMAL CONDITION PIE CHART

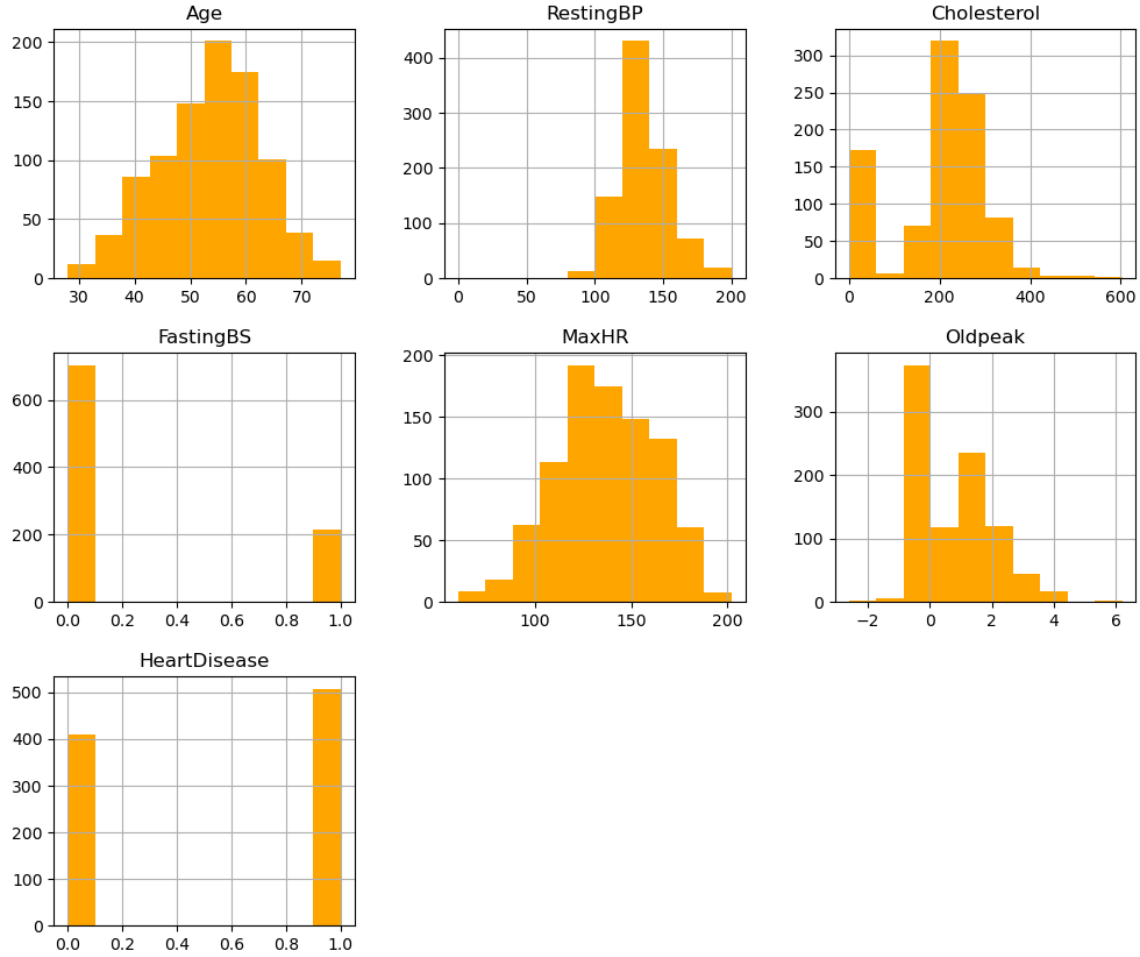
In [9]:

```
labels=['HEART DISEASED','NORMAL']
count= data['HeartDisease'].value_counts()
plt.figure(figsize=(8,8))
plt.pie(count,labels=labels,explode=(0,.1), colors=['red','grey'])
plt.legend( ['heart disease','Normal'],loc =1)

plt.title('Heart Disease', fontsize= 40)
plt.show()
```



HISTOGRAMS REPRESENTING ALL COLUMNS

In [10]:
```
# HISTOGRAMS REPRESENTING ALL COLUMNS
data.hist(figsize=(12,10), color='orange')
plt.show()
```



In [40]:
```
# DIVIDE DATA ACCORDING TO DATA TYPES
obj = data.select_dtypes(include='object')
not_obj = data.select_dtypes(exclude='object')
```

In [24]:
```
not_obj
```

Out[24]:

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
0	40	140	289	0	172	0.0	0
1	49	160	180	0	156	1.0	1
2	37	130	283	0	98	0.0	0
3	48	138	214	0	108	1.5	1
4	54	150	195	0	122	0.0	0
...	...	...	...	...	...	...	...
913	45	110	264	0	132	1.2	1
914	68	144	193	1	141	3.4	1
915	57	130	131	0	115	1.2	1
916	57	130	236	0	174	0.0	1
917	38	138	175	0	173	0.0	0

918 rows × 7 columns

REPRESENT CORRELATION OF DATA

In [25]:
```
# REPRESENT CORRELATION OF DATA
corr=not_obj.corr()
corr
```

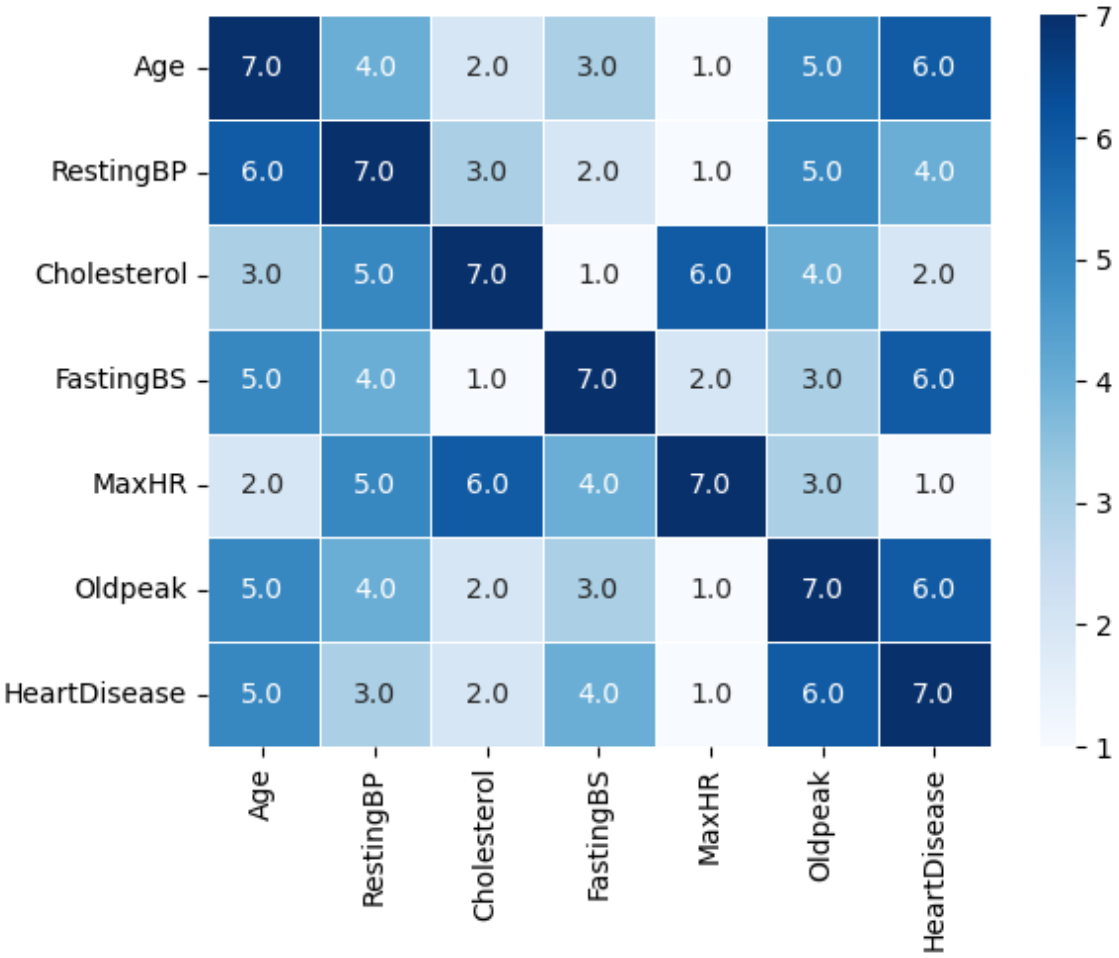
Out[25]:

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDiseas
Age	1.000000	0.254399	-0.095282	0.198039	-0.382045	0.258612	0.28203
RestingBP	0.254399	1.000000	0.100893	0.070193	-0.112135	0.164803	0.10758
Cholesterol	-0.095282	0.100893	1.000000	-0.260974	0.235792	0.050148	-0.23274
FastingBS	0.198039	0.070193	-0.260974	1.000000	-0.131438	0.052698	0.26729
MaxHR	-0.382045	-0.112135	0.235792	-0.131438	1.000000	-0.160691	-0.40042
Oldpeak	0.258612	0.164803	0.050148	0.052698	-0.160691	1.000000	0.40395
HeartDisease	0.282039	0.107589	-0.232741	0.267291	-0.400421	0.403951	1.00000

HEATMAP FOR CORRELATION BETWEEN VARIABLES

In [28]:
```
# HEATMAP FOR CORRELATION BETWEEN VARIABLES
import seaborn as sns
sns.heatmap(corr.rank(axis='columns'), annot=True,fmt='.1f',linewidth=.5, c
```

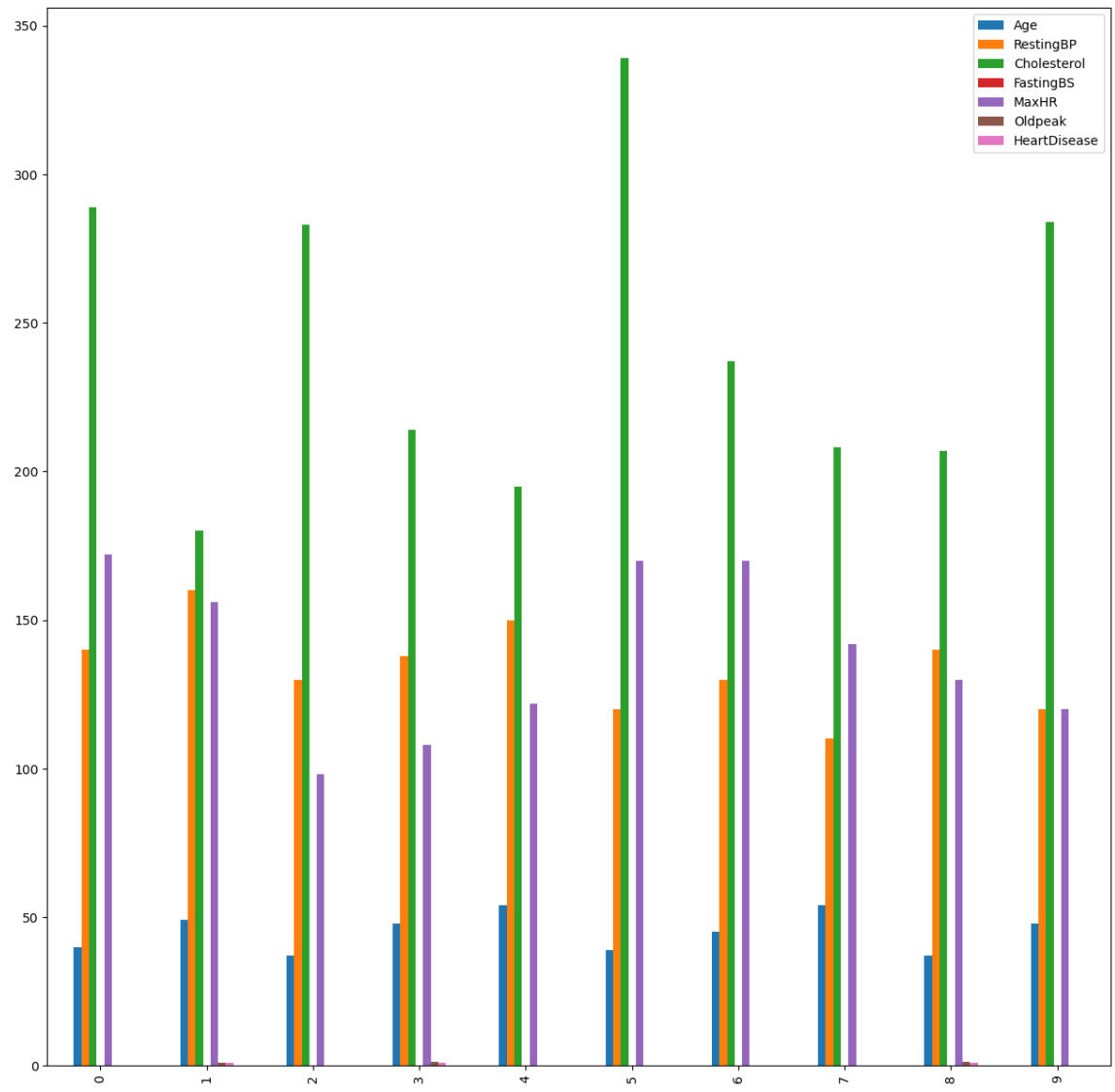
Out[28]: <Axes: >



BAR GRAPH REPRESENTING ALL COLUMNS

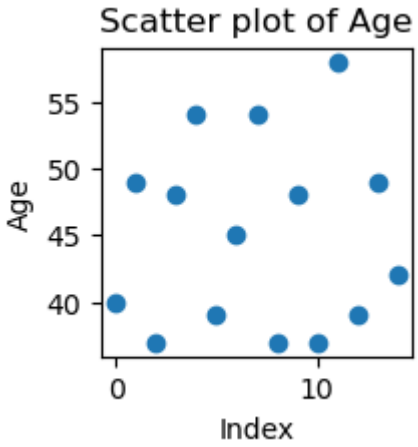
In [46]:
```
# BAR GRAPH REPRESENTING ALL COLUMNS
new= not_obj.head(n=10)
new.plot(kind='bar',figsize=(15,15))
plt.title('BAR GRAPH REPRESENTS ALL COLUMNS', fontsize = 40, pad= 40)
```

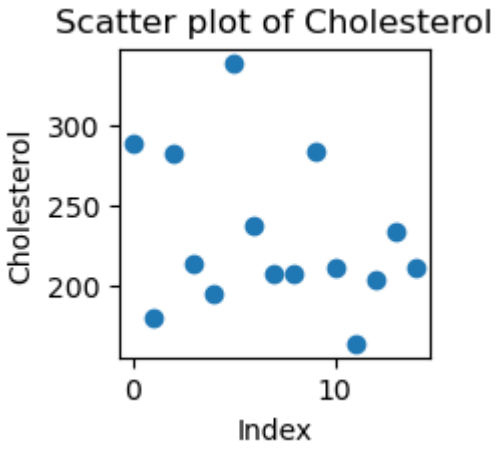
Out[46]: Text(0.5, 1.0, 'BAR GRAPH REPRESENTS ALL COLUMNS')

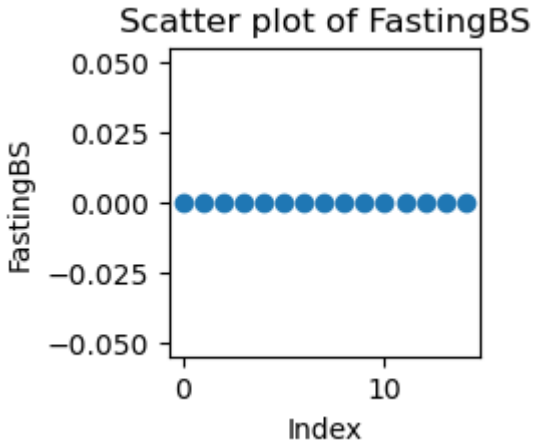


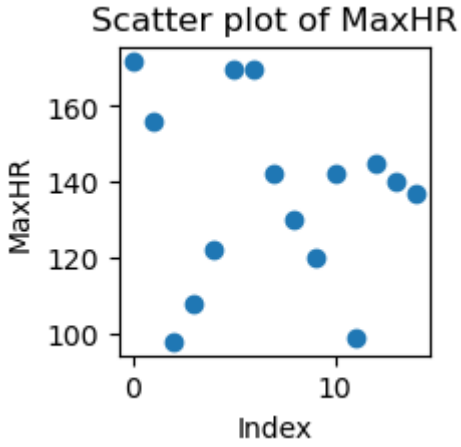
SCATTER PLOT FOR NUMERICAL COLUMN

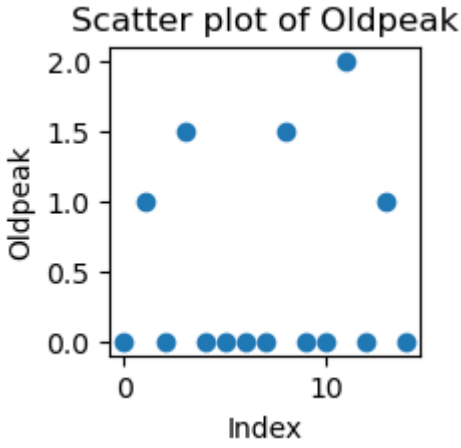
In [14]:
```
# scatter plot for every column

new_data = data.head(n=15)
numerical_columns = new_data.select_dtypes(include=['int64', 'float64']).co
for column in numerical_columns:
    plt.figure(figsize=(2, 2))
    plt.scatter(new_data.index, new_data[column])
    plt.title(f'Scatter plot of {column}')
    plt.xlabel('Index')
    plt.ylabel(column)
    plt.show()
```

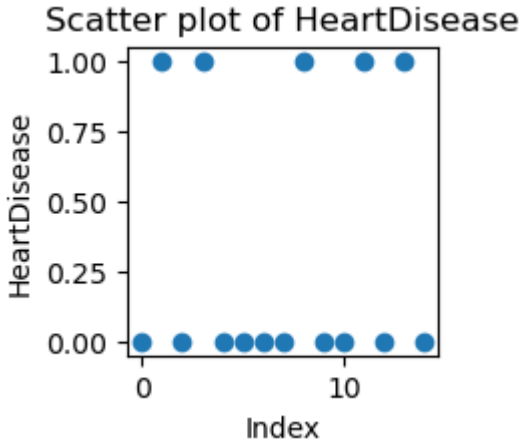





Scatter plot of FastingBS
FastingBS
0.050
0.025
0.000
−0.025
−0.050
0
10
Index

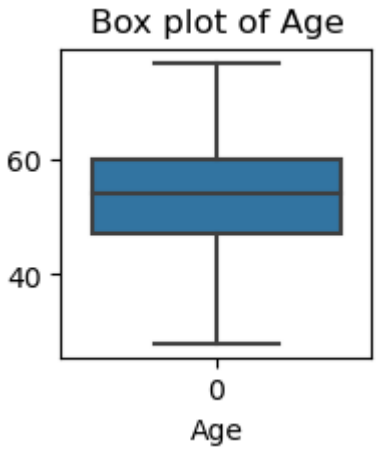

Scatter plot of MaxHR
MaxHR
160
140
120
100
0
10
Index

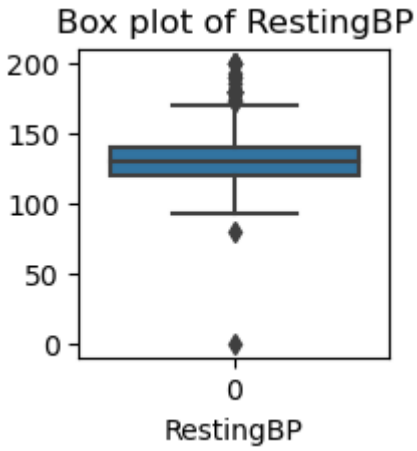

Scatter plot of Oldpeak
Oldpeak
2.0
1.5
1.0
0.5
0.0
0
10
Index

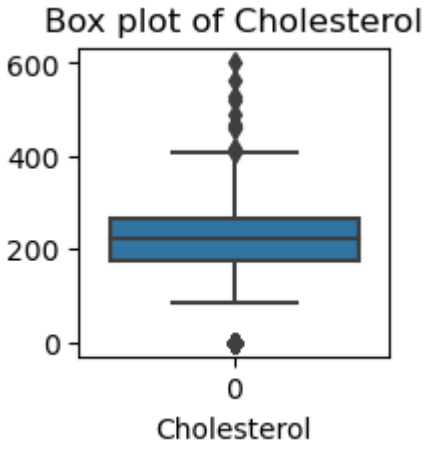

Scatter plot of HeartDisease
HeartDisease
1.00
0.75
0.50
0.25
0.00
0
10
Index

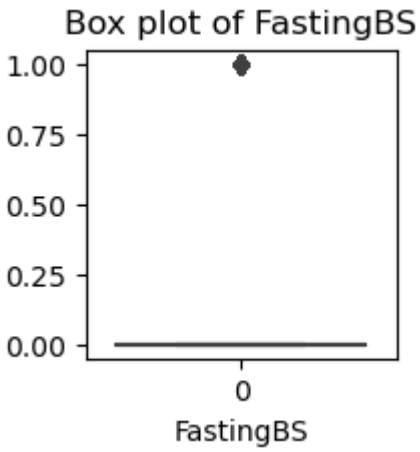
BOX PLOT FOR NUMERICAL COLUMNS

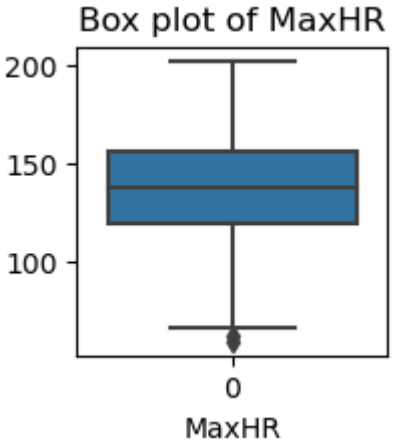
In [16]:
```
# BOX PLOT FOR NUMERICAL COLUMNS
import seaborn as sns
for column in numerical_columns:
    plt.figure(figsize=(2, 2))
    sns.boxplot(data[column])
    plt.title(f'Box plot of {column}')
    plt.xlabel(column)
    plt.show()
```

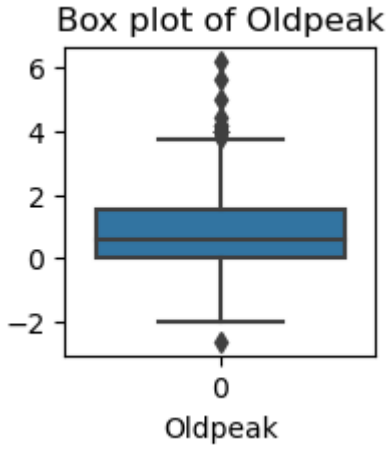


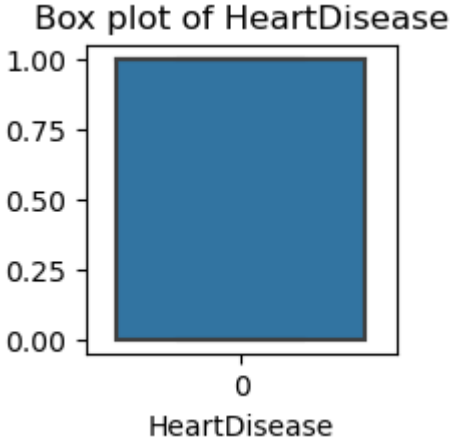





Box plot of FastingBS
1.00
0.75
0.50
0.25
0.00
0
FastingBS


Box plot of MaxHR
200
150
100
0
MaxHR


Box plot of Oldpeak
6
4
2
0
−2
0
Oldpeak


Box plot of HeartDisease
1.00
0.75
0.50
0.25
0.00
0
HeartDisease