In [39]:
```
# importing all necessary libraries
# pip install pandas
# pip install seaborn
# pip install numpy
# pip install matplotlib
# pip install scikit-learn
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings("ignore", category=UserWarning, module='sklearn')
```

In [3]:
```
data = pd.read_csv('D:\\data_with_diseases.csv')
```

In [4]:
```
data
```

Out[4]:

	Sl. No	Patient File No.	PCOS (Y/N)	Age (yrs)	Weight (Kg)	Height(Cm)	BMI	Blood Group	Pulse rate(bpm)	(breat
0	1	1	0	28	44.6	152.000	19.300000	15	78	
1	2	2	0	36	65.0	161.500	24.921163	15	74	
2	3	3	1	33	68.8	165.000	25.270891	11	72	
3	4	4	0	37	65.0	148.000	29.674945	13	72	
4	5	5	0	25	52.0	161.000	20.060954	11	72	
...	...	...	...	...	...	...	...	...	...	
536	537	537	0	35	50.0	164.592	18.500000	17	72	
537	538	538	0	30	63.2	158.000	25.300000	15	72	
538	539	539	0	36	54.0	152.000	23.400000	13	74	
539	540	540	0	27	50.0	150.000	22.200000	15	74	
540	541	541	1	23	82.0	165.000	30.100000	13	80	

541 rows × 46 columns

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541 entries, 0 to 540
Data columns (total 46 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Sl. No                  541 non-null    int64  
 1   Patient File No.        541 non-null    int64  
 2   PCOS (Y/N)              541 non-null    int64  
 3    Age (yrs)              541 non-null    int64  
 4   Weight (Kg)             541 non-null    float64
 5   Height(Cm)              541 non-null    float64
 6   BMI                     541 non-null    float64
 7   Blood Group             541 non-null    int64  
 8   Pulse rate(bpm)         541 non-null    int64  
 9   RR (breaths/min)        541 non-null    int64  
 10  Hb(g/dl)                541 non-null    float64
 11  Cycle(R/I)              541 non-null    int64  
 12  Cycle length(days)      541 non-null    int64  
 13  Marraige Status (Yrs)   540 non-null    float64
 14  Pregnant(Y/N)           541 non-null    int64  
 15  No. of aborptions       541 non-null    int64  
 16    I   beta-HCG(mIU/mL)  541 non-null    float64
 17  II    beta-HCG(mIU/mL)  541 non-null    float64
 18  Hip(inch)               541 non-null    int64  
 19  Waist(inch)             541 non-null    int64  
 20  Waist:Hip Ratio         541 non-null    float64
 21  TSH (mIU/L)             541 non-null    float64
 22  AMH(ng/mL)              541 non-null    object 
 23  PRL(ng/mL)              541 non-null    float64
 24  Vit D3 (ng/mL)          541 non-null    float64
 25  PRG(ng/mL)              541 non-null    float64
 26  RBS(mg/dl)              541 non-null    float64
 27  Weight gain(Y/N)        541 non-null    int64  
 28  hair growth(Y/N)        541 non-null    int64  
 29  Skin darkening (Y/N)    541 non-null    int64  
 30  Hair loss(Y/N)          541 non-null    int64  
 31  Pimples(Y/N)            541 non-null    int64  
 32  Fast food (Y/N)         540 non-null    float64
 33  Reg.Exercise(Y/N)       541 non-null    int64  
 34  BP _Systolic (mmHg)     541 non-null    int64  
 35  BP _Diastolic (mmHg)    541 non-null    int64  
 36  Follicle No. (L)        541 non-null    int64  
 37  Follicle No. (R)        541 non-null    int64  
 38  Avg. F size (L) (mm)    541 non-null    float64
 39  Avg. F size (R) (mm)    541 non-null    float64
 40  Endometrium (mm)        541 non-null    float64
 41  Unnamed: 41             2 non-null      object 
 42  Diabetes                541 non-null    int64  
 43  Hypertension            541 non-null    int64  
 44  Infertility             541 non-null    int64  
 45  stressLevel             541 non-null    int64  
dtypes: float64(17), int64(27), object(2)
memory usage: 194.6+ KB
```

In [7]:
```
data.head()
```

Out[7]:

	Sl. No	Patient File No.	PCOS (Y/N)	Age (yrs)	Weight (Kg)	Height(Cm)	BMI	Blood Group	Pulse rate(bpm)	(breaths/
0	1	1	0	28	44.6	152.0	19.300000	15	78	
1	2	2	0	36	65.0	161.5	24.921163	15	74	
2	3	3	1	33	68.8	165.0	25.270891	11	72	
3	4	4	0	37	65.0	148.0	29.674945	13	72	
4	5	5	0	25	52.0	161.0	20.060954	11	72	

5 rows × 46 columns

In [8]:
```
data.describe()
```

Out[8]:

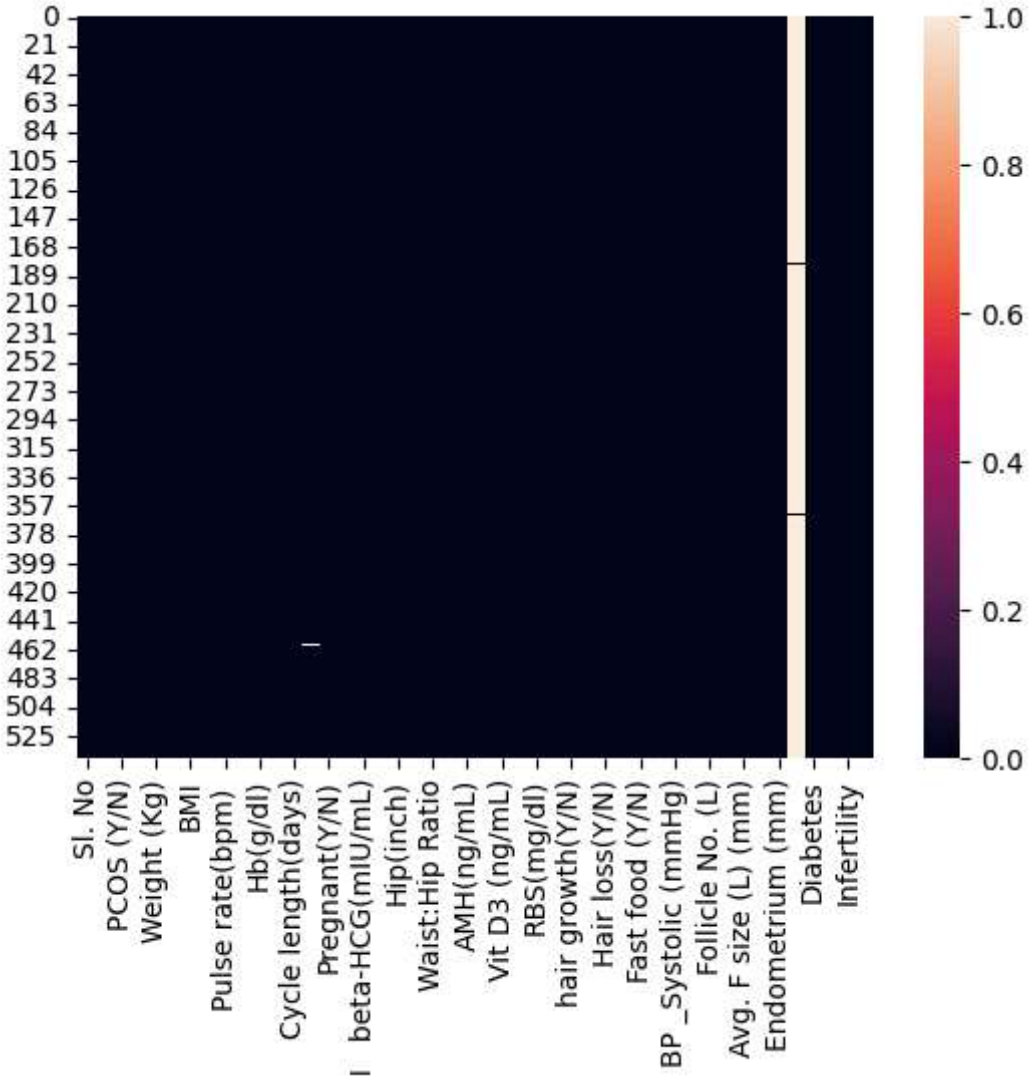
	Sl. No	Patient File No.	PCOS (Y/N)	Age (yrs)	Weight (Kg)	Height(Cm)	BMI
count	541.000000	541.000000	541.000000	541.000000	541.000000	541.000000	541.000000
mean	271.000000	271.000000	0.327172	31.430684	59.637153	156.484835	24.311224
std	156.317519	156.317519	0.469615	5.411006	11.028287	6.033545	4.056195
min	1.000000	1.000000	0.000000	20.000000	31.000000	137.000000	12.417882
25%	136.000000	136.000000	0.000000	28.000000	52.000000	152.000000	21.641274
50%	271.000000	271.000000	0.000000	31.000000	59.000000	156.000000	24.238227
75%	406.000000	406.000000	1.000000	35.000000	65.000000	160.000000	26.634958
max	541.000000	541.000000	1.000000	48.000000	108.000000	180.000000	38.900000

8 rows × 44 columns

Checking for empty values

In [9]:
```
sns.heatmap(data.isnull())
```

Out[9]:
```
<Axes: >
```



Droping the unnecesary colums(parameters)

In [10]:
```
data.drop(["Marraige Status (Yrs)","Fast food (Y/N)","Unnamed: 41","Sl. No","Patien
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541 entries, 0 to 540
Data columns (total 41 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   PCOS (Y/N)              541 non-null    int64  
 1    Age (yrs)              541 non-null    int64  
 2   Weight (Kg)             541 non-null    float64
 3   Height(Cm)              541 non-null    float64
 4   BMI                     541 non-null    float64
 5   Blood Group             541 non-null    int64  
 6   Pulse rate(bpm)         541 non-null    int64  
 7   RR (breaths/min)        541 non-null    int64  
 8   Hb(g/dl)                541 non-null    float64
 9   Cycle(R/I)              541 non-null    int64  
 10  Cycle length(days)      541 non-null    int64  
 11  Pregnant(Y/N)           541 non-null    int64  
 12  No. of aborptions       541 non-null    int64  
 13    I   beta-HCG(mIU/mL)  541 non-null    float64
 14  II    beta-HCG(mIU/mL)  541 non-null    float64
 15  Hip(inch)               541 non-null    int64  
 16  Waist(inch)             541 non-null    int64  
 17  Waist:Hip Ratio         541 non-null    float64
 18  TSH (mIU/L)             541 non-null    float64
 19  AMH(ng/mL)              541 non-null    object 
 20  PRL(ng/mL)              541 non-null    float64
 21  Vit D3 (ng/mL)          541 non-null    float64
 22  PRG(ng/mL)              541 non-null    float64
 23  RBS(mg/dl)              541 non-null    float64
 24  Weight gain(Y/N)        541 non-null    int64  
 25  hair growth(Y/N)        541 non-null    int64  
 26  Skin darkening (Y/N)    541 non-null    int64  
 27  Hair loss(Y/N)          541 non-null    int64  
 28  Pimples(Y/N)            541 non-null    int64  
 29  Reg.Exercise(Y/N)       541 non-null    int64  
 30  BP _Systolic (mmHg)     541 non-null    int64  
 31  BP _Diastolic (mmHg)    541 non-null    int64  
 32  Follicle No. (L)        541 non-null    int64  
 33  Follicle No. (R)        541 non-null    int64  
 34  Avg. F size (L) (mm)    541 non-null    float64
 35  Avg. F size (R) (mm)    541 non-null    float64
 36  Endometrium (mm)        541 non-null    float64
 37  Diabetes                541 non-null    int64  
 38  Hypertension            541 non-null    int64  
 39  Infertility             541 non-null    int64  
 40  stressLevel             541 non-null    int64  
dtypes: float64(15), int64(25), object(1)
memory usage: 173.4+ KB
```

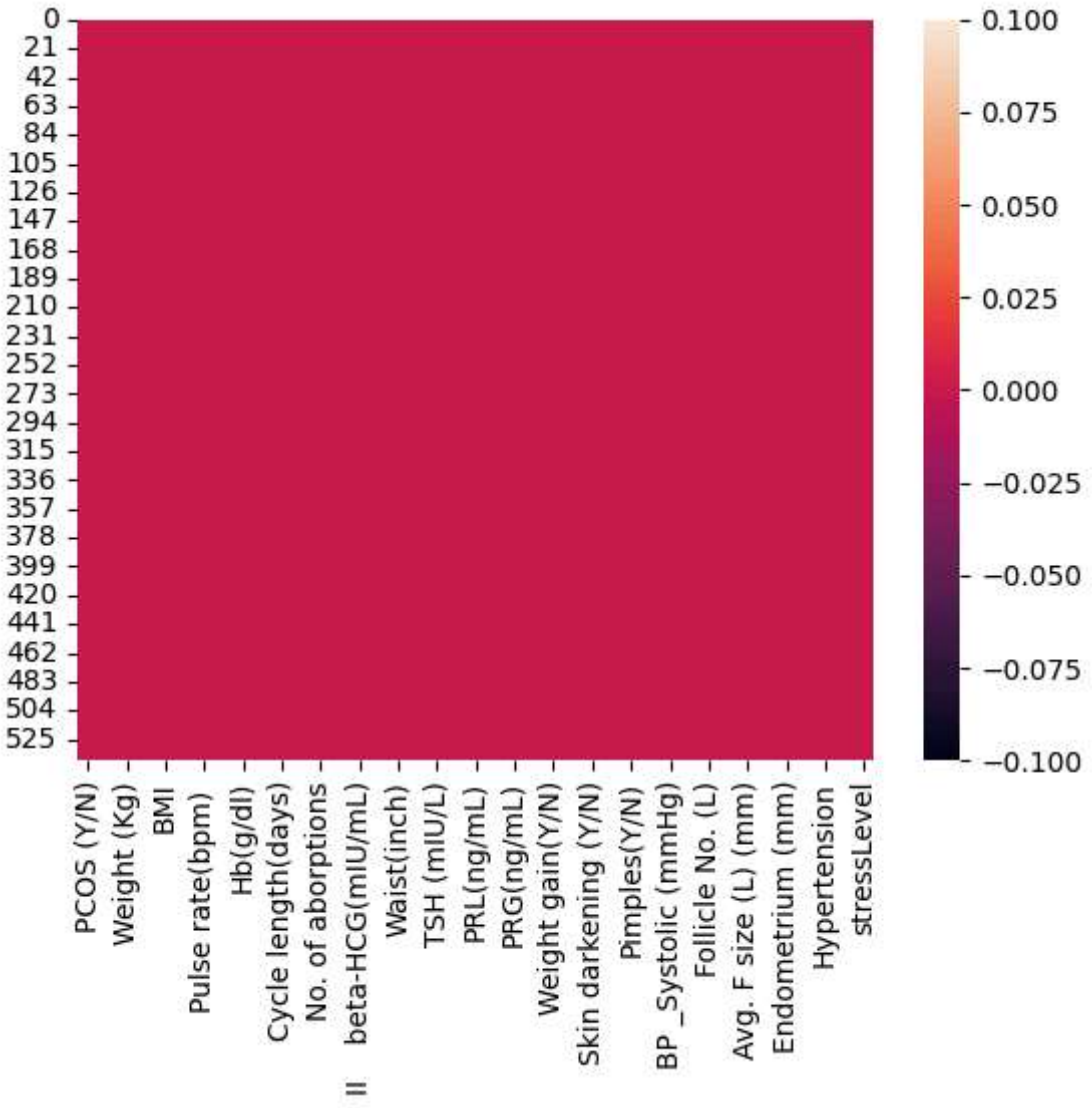
Conforming no empty values

In [11]:
```
data.isnull().sum()
```

Out[11]:
```
PCOS (Y/N)                0
 Age (yrs)                0
Weight (Kg)               0
Height(Cm)                0
BMI                       0
Blood Group               0
Pulse rate(bpm)           0
RR (breaths/min)          0
Hb(g/dl)                  0
Cycle(R/I)                0
Cycle length(days)        0
Pregnant(Y/N)             0
No. of aborptions         0
  I   beta-HCG(mIU/mL)    0
II    beta-HCG(mIU/mL)    0
Hip(inch)                 0
Waist(inch)               0
Waist:Hip Ratio           0
TSH (mIU/L)               0
AMH(ng/mL)                0
PRL(ng/mL)                0
Vit D3 (ng/mL)            0
PRG(ng/mL)                0
RBS(mg/dl)                0
Weight gain(Y/N)          0
hair growth(Y/N)          0
Skin darkening (Y/N)      0
Hair loss(Y/N)            0
Pimples(Y/N)              0
Reg.Exercise(Y/N)         0
BP _Systolic (mmHg)       0
BP _Diastolic (mmHg)      0
Follicle No. (L)          0
Follicle No. (R)          0
Avg. F size (L) (mm)      0
Avg. F size (R) (mm)      0
Endometrium (mm)          0
Diabetes                  0
Hypertension              0
Infertility               0
stressLevel               0
dtype: int64
```

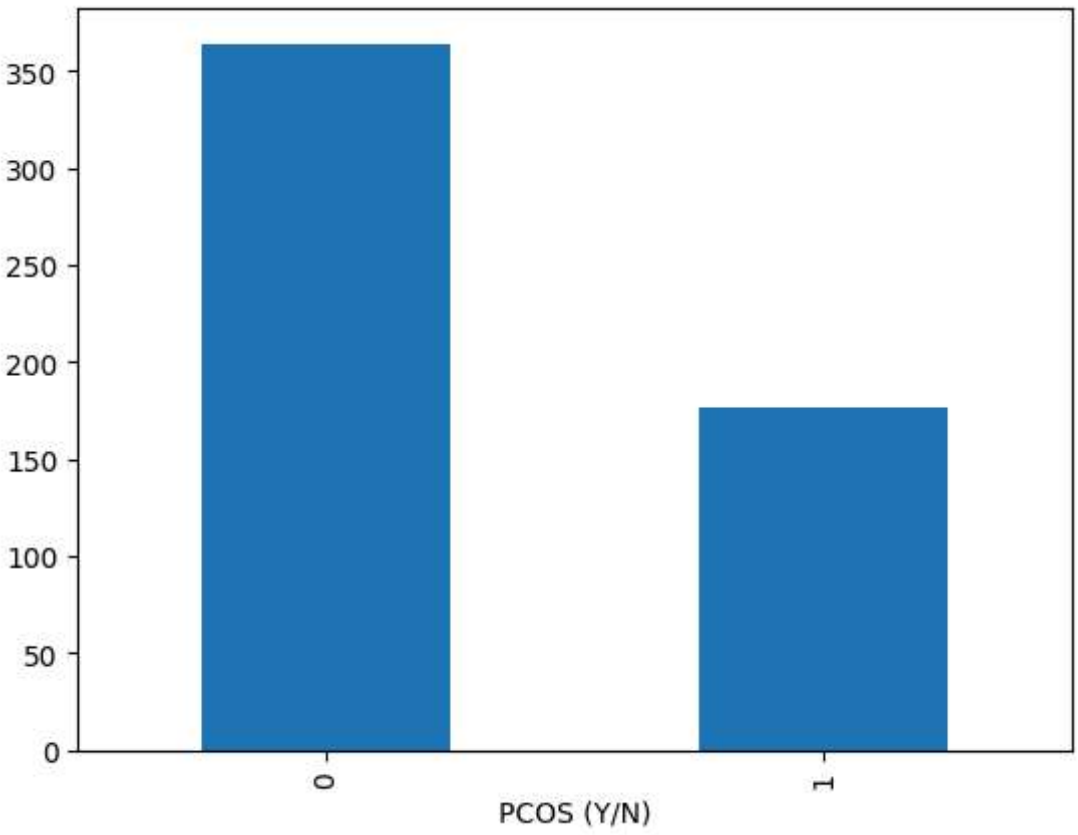
In [12]:
```
sns.heatmap(data.isnull())
```

Out[12]:
```
<Axes: >
```



In [13]:
```
data["PCOS (Y/N)"].value_counts().plot(kind = "bar")
```

Out[13]:
```
<Axes: xlabel='PCOS (Y/N)'>
```



In [14]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541 entries, 0 to 540
Data columns (total 41 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   PCOS (Y/N)              541 non-null    int64  
 1    Age (yrs)              541 non-null    int64  
 2   Weight (Kg)             541 non-null    float64
 3   Height(Cm)              541 non-null    float64
 4   BMI                     541 non-null    float64
 5   Blood Group             541 non-null    int64  
 6   Pulse rate(bpm)         541 non-null    int64  
 7   RR (breaths/min)        541 non-null    int64  
 8   Hb(g/dl)                541 non-null    float64
 9   Cycle(R/I)              541 non-null    int64  
 10  Cycle length(days)      541 non-null    int64  
 11  Pregnant(Y/N)           541 non-null    int64  
 12  No. of aborptions       541 non-null    int64  
 13    I   beta-HCG(mIU/mL)  541 non-null    float64
 14  II    beta-HCG(mIU/mL)  541 non-null    float64
 15  Hip(inch)               541 non-null    int64  
 16  Waist(inch)             541 non-null    int64  
 17  Waist:Hip Ratio         541 non-null    float64
 18  TSH (mIU/L)             541 non-null    float64
 19  AMH(ng/mL)              541 non-null    object 
 20  PRL(ng/mL)              541 non-null    float64
 21  Vit D3 (ng/mL)          541 non-null    float64
 22  PRG(ng/mL)              541 non-null    float64
 23  RBS(mg/dl)              541 non-null    float64
 24  Weight gain(Y/N)        541 non-null    int64  
 25  hair growth(Y/N)        541 non-null    int64  
 26  Skin darkening (Y/N)    541 non-null    int64  
 27  Hair loss(Y/N)          541 non-null    int64  
 28  Pimples(Y/N)            541 non-null    int64  
 29  Reg.Exercise(Y/N)       541 non-null    int64  
 30  BP _Systolic (mmHg)     541 non-null    int64  
 31  BP _Diastolic (mmHg)    541 non-null    int64  
 32  Follicle No. (L)        541 non-null    int64  
 33  Follicle No. (R)        541 non-null    int64  
 34  Avg. F size (L) (mm)    541 non-null    float64
 35  Avg. F size (R) (mm)    541 non-null    float64
 36  Endometrium (mm)        541 non-null    float64
 37  Diabetes                541 non-null    int64  
 38  Hypertension            541 non-null    int64  
 39  Infertility             541 non-null    int64  
 40  stressLevel             541 non-null    int64  
dtypes: float64(15), int64(25), object(1)
memory usage: 173.4+ KB
```

Converting all data to float type

In [15]:
```
for col in data.select_dtypes(include='int64'):  # Adjust dtype as needed, e.g., 'i
    data[col] = data[col].astype(float)
```

In [16]:
```
data["AMH(ng/mL)"] = pd.to_numeric(data["AMH(ng/mL)"], errors='coerce')
data.fillna({'AMH(ng/mL)' : data['AMH(ng/mL)'].median()} , inplace = True)
```

In [17]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541 entries, 0 to 540
Data columns (total 41 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   PCOS (Y/N)              541 non-null    float64
 1    Age (yrs)              541 non-null    float64
 2   Weight (Kg)             541 non-null    float64
 3   Height(Cm)              541 non-null    float64
 4   BMI                     541 non-null    float64
 5   Blood Group             541 non-null    float64
 6   Pulse rate(bpm)         541 non-null    float64
 7   RR (breaths/min)        541 non-null    float64
 8   Hb(g/dl)                541 non-null    float64
 9   Cycle(R/I)              541 non-null    float64
 10  Cycle length(days)      541 non-null    float64
 11  Pregnant(Y/N)           541 non-null    float64
 12  No. of aborptions       541 non-null    float64
 13    I   beta-HCG(mIU/mL)  541 non-null    float64
 14  II    beta-HCG(mIU/mL)  541 non-null    float64
 15  Hip(inch)               541 non-null    float64
 16  Waist(inch)             541 non-null    float64
 17  Waist:Hip Ratio         541 non-null    float64
 18  TSH (mIU/L)             541 non-null    float64
 19  AMH(ng/mL)              541 non-null    float64
 20  PRL(ng/mL)              541 non-null    float64
 21  Vit D3 (ng/mL)          541 non-null    float64
 22  PRG(ng/mL)              541 non-null    float64
 23  RBS(mg/dl)              541 non-null    float64
 24  Weight gain(Y/N)        541 non-null    float64
 25  hair growth(Y/N)        541 non-null    float64
 26  Skin darkening (Y/N)    541 non-null    float64
 27  Hair loss(Y/N)          541 non-null    float64
 28  Pimples(Y/N)            541 non-null    float64
 29  Reg.Exercise(Y/N)       541 non-null    float64
 30  BP _Systolic (mmHg)     541 non-null    float64
 31  BP _Diastolic (mmHg)    541 non-null    float64
 32  Follicle No. (L)        541 non-null    float64
 33  Follicle No. (R)        541 non-null    float64
 34  Avg. F size (L) (mm)    541 non-null    float64
 35  Avg. F size (R) (mm)    541 non-null    float64
 36  Endometrium (mm)        541 non-null    float64
 37  Diabetes                541 non-null    float64
 38  Hypertension            541 non-null    float64
 39  Infertility             541 non-null    float64
 40  stressLevel             541 non-null    float64
dtypes: float64(41)
memory usage: 173.4 KB
```

In [18]:
```
data['PCOS (Y/N)'] = data['PCOS (Y/N)'].astype(int)
```

In [19]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541 entries, 0 to 540
Data columns (total 41 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   PCOS (Y/N)              541 non-null    int64  
 1    Age (yrs)              541 non-null    float64
 2   Weight (Kg)             541 non-null    float64
 3   Height(Cm)              541 non-null    float64
 4   BMI                     541 non-null    float64
 5   Blood Group             541 non-null    float64
 6   Pulse rate(bpm)         541 non-null    float64
 7   RR (breaths/min)        541 non-null    float64
 8   Hb(g/dl)                541 non-null    float64
 9   Cycle(R/I)              541 non-null    float64
 10  Cycle length(days)      541 non-null    float64
 11  Pregnant(Y/N)           541 non-null    float64
 12  No. of aborptions       541 non-null    float64
 13    I   beta-HCG(mIU/mL)  541 non-null    float64
 14  II    beta-HCG(mIU/mL)  541 non-null    float64
 15  Hip(inch)               541 non-null    float64
 16  Waist(inch)             541 non-null    float64
 17  Waist:Hip Ratio         541 non-null    float64
 18  TSH (mIU/L)             541 non-null    float64
 19  AMH(ng/mL)              541 non-null    float64
 20  PRL(ng/mL)              541 non-null    float64
 21  Vit D3 (ng/mL)          541 non-null    float64
 22  PRG(ng/mL)              541 non-null    float64
 23  RBS(mg/dl)              541 non-null    float64
 24  Weight gain(Y/N)        541 non-null    float64
 25  hair growth(Y/N)        541 non-null    float64
 26  Skin darkening (Y/N)    541 non-null    float64
 27  Hair loss(Y/N)          541 non-null    float64
 28  Pimples(Y/N)            541 non-null    float64
 29  Reg.Exercise(Y/N)       541 non-null    float64
 30  BP _Systolic (mmHg)     541 non-null    float64
 31  BP _Diastolic (mmHg)    541 non-null    float64
 32  Follicle No. (L)        541 non-null    float64
 33  Follicle No. (R)        541 non-null    float64
 34  Avg. F size (L) (mm)    541 non-null    float64
 35  Avg. F size (R) (mm)    541 non-null    float64
 36  Endometrium (mm)        541 non-null    float64
 37  Diabetes                541 non-null    float64
 38  Hypertension            541 non-null    float64
 39  Infertility             541 non-null    float64
 40  stressLevel             541 non-null    float64
dtypes: float64(40), int64(1)
memory usage: 173.4 KB
```

Dependent and independent variables selection

In [20]:
```
y = data["PCOS (Y/N)"]
x = data.drop(["PCOS (Y/N)"], axis = 1)
```

In [21]:
```
x
```

Out[21]:

	Age (yrs)	Weight (Kg)	Height(Cm)	BMI	Blood Group	Pulse rate(bpm)	RR (breaths/min)	Hb(g/dl)	Cy
0	28.0	44.6	152.000	19.300000	15.0	78.0	22.0	10.48	
1	36.0	65.0	161.500	24.921163	15.0	74.0	20.0	11.70	
2	33.0	68.8	165.000	25.270891	11.0	72.0	18.0	11.80	
3	37.0	65.0	148.000	29.674945	13.0	72.0	20.0	12.00	
4	25.0	52.0	161.000	20.060954	11.0	72.0	18.0	10.00	
...	...	...	...	...	...	...	...	...	
536	35.0	50.0	164.592	18.500000	17.0	72.0	16.0	11.00	
537	30.0	63.2	158.000	25.300000	15.0	72.0	18.0	10.80	
538	36.0	54.0	152.000	23.400000	13.0	74.0	20.0	10.80	
539	27.0	50.0	150.000	22.200000	15.0	74.0	20.0	12.00	
540	23.0	82.0	165.000	30.100000	13.0	80.0	20.0	10.20	

541 rows × 40 columns

In [22]:
```
y
```

Out[22]:
```
0      0
1      0
2      1
3      0
4      0
      ..
536    0
537    0
538    0
539    0
540    1
Name: PCOS (Y/N), Length: 541, dtype: int64
```

In [23]:
```
data.head()
```

Out[23]:

	PCOS (Y/N)	Age (yrs)	Weight (Kg)	Height(Cm)	BMI	Blood Group	Pulse rate(bpm)	RR (breaths/min)	Hb(g/dl
0	0	28.0	44.6	152.0	19.300000	15.0	78.0	22.0	10.4
1	0	36.0	65.0	161.5	24.921163	15.0	74.0	20.0	11.7
2	1	33.0	68.8	165.0	25.270891	11.0	72.0	18.0	11.8
3	0	37.0	65.0	148.0	29.674945	13.0	72.0	20.0	12.0
4	0	25.0	52.0	161.0	20.060954	11.0	72.0	18.0	10.0

5 rows × 41 columns

Normalize the data (using scikit-learn library)

In [24]:
```
scaler = StandardScaler()
x_scaler = scaler.fit_transform(x)
```

In [25]:
```
x_scaler
```

Out[25]:
```
array([[-0.63460643, -1.36476953, -0.74400482, ..., -0.6798251 ,
        -0.39181389, -0.97452985],
       [ 0.84523012,  0.48673106,  0.83198292, ..., -0.6798251 ,
        -0.39181389, -0.20710532],
       [ 0.29029141,  0.83161842,  1.41260999, ...,  1.47096658,
        -0.39181389,  0.5603192 ],
       ...,
       [ 0.84523012, -0.5116271 , -0.74400482, ..., -0.6798251 ,
        -0.39181389, -0.20710532],
       [-0.819586  , -0.87466643, -1.07579171, ..., -0.6798251 ,
        -0.39181389, -0.97452985],
       [-1.55950427,  2.02964822,  1.41260999, ...,  1.47096658,
         2.55223214,  0.5603192 ]], shape=(541, 40))
```

Split the data

In [26]:
```
x_train,x_test,y_train,y_test = train_test_split(x , y , test_size = 0.30 , random_
```

In [27]:
```
print("X_train",len(x_train))
print("X_test",len(x_test))
print("y_train",len(y_train))
print("y_test",len(y_test))
```

```
X_train 378
X_test 163
y_train 378
y_test 163
```

Import logistic regression and Train the model

In [28]:
```
lr = LogisticRegression()
lr.fit(x_train, y_train)
acc_log_train = round(lr.score(x_train, y_train)*100,2)
acc_log_test = round(lr.score(x_test,y_test)*100,2)
print(f"Training Accuracy: {acc_log_train} %")
print(f"Training Accuracy: {acc_log_test} %")
```

```
Training Accuracy: 93.39 %
Training Accuracy: 88.96 %
```

In [29]:
```
#creat the model
lr = LogisticRegression()
lr.fit(x_train,y_train)

#predict the target variable
y_prediction_values = lr.predict(x_test)
```

In [30]:
```
y_prediction_values
```

Out[30]:
```
array([0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0])
```

In [31]:
```
y_test
```

Out[31]:
```
229    0
73     0
352    1
86     0
470    1
      ..
275    1
362    0
31     1
113    1
381    0
Name: PCOS (Y/N), Length: 163, dtype: int64
```

Checking the accuracy of the model

In [32]:
```
#accuracy score
accuracy = accuracy_score(y_test,y_prediction_values)
print(f"Accuracy: {accuracy: 2f}")
```

```
Accuracy:  0.889571
```

In [33]:
```
#classification report
print(classification_report(y_test,y_prediction_values))
```

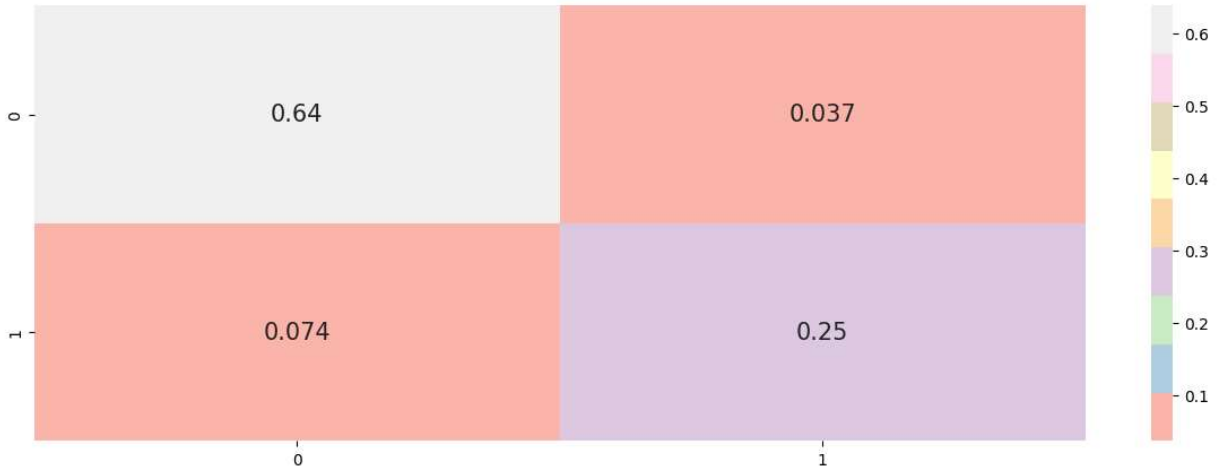
```
              precision    recall  f1-score   support

           0       0.90      0.95      0.92       110
           1       0.87      0.77      0.82        53

    accuracy                           0.89       163
   macro avg       0.88      0.86      0.87       163
weighted avg       0.89      0.89      0.89       163
```

In [34]:
```
y_prob = lr.predict_proba(x_test)
print("Predicted probabilities:", y_prob[:5])
```

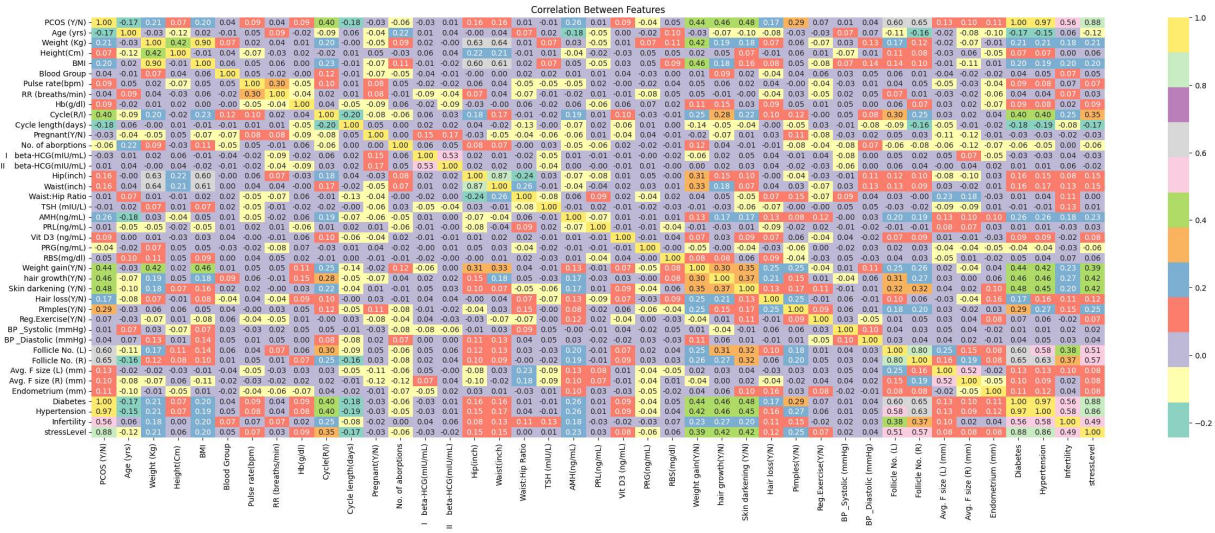
```
Predicted probabilities: [[0.78741781 0.21258219]
 [0.96176421 0.03823579]
 [0.93276185 0.06723815]
 [0.44142847 0.55857153]
 [0.20652848 0.79347152]]
```

In [35]:
```
# cofusion matrix
from sklearn.metrics import confusion_matrix
plt.subplots(figsize=(15,5))
cf_matrix = confusion_matrix(y_test, y_prediction_values)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot = True, annot_kws = {'size':15}, cma
```

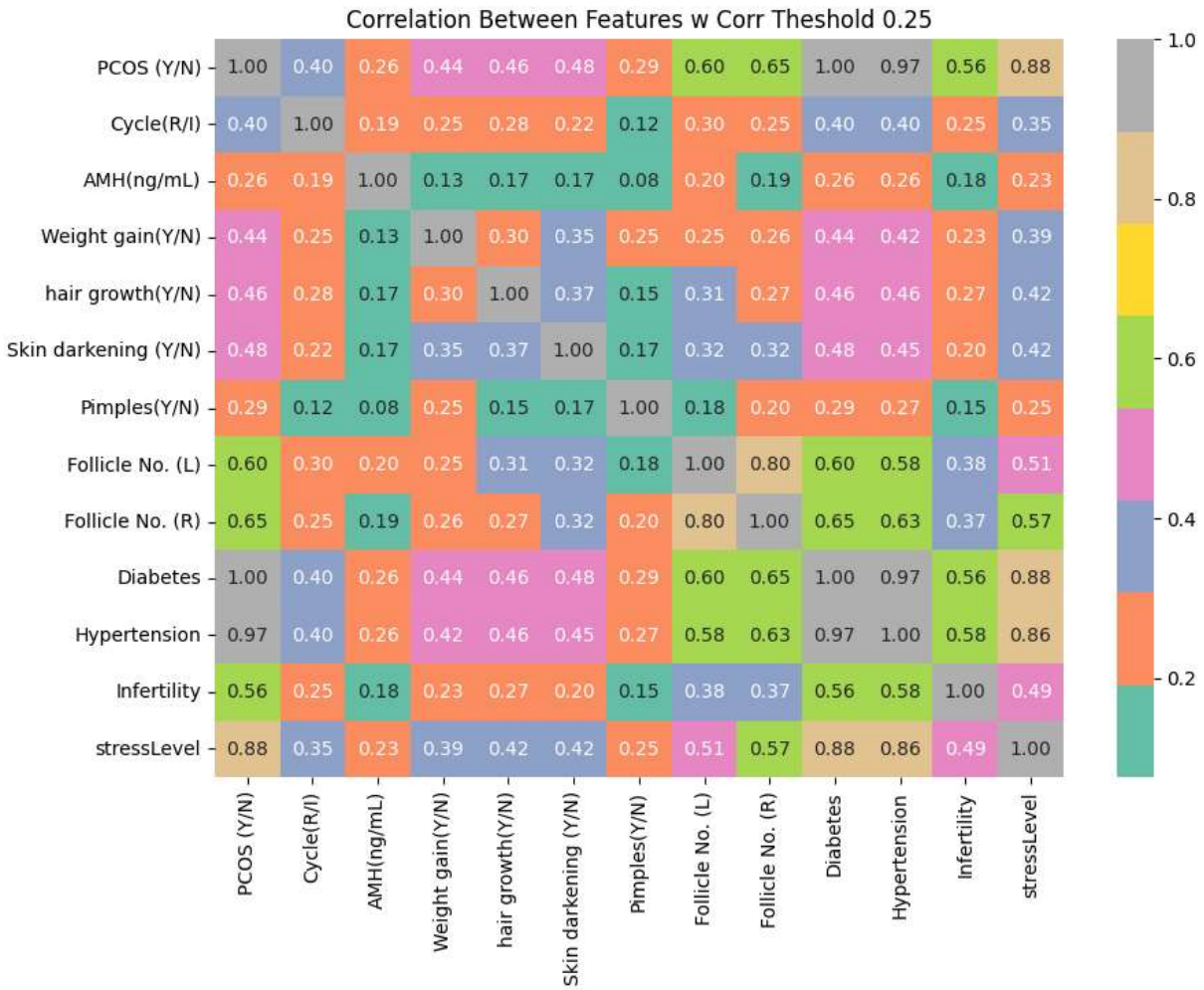
Out[35]: <Axes: >



In [36]:
```
corr_matrix= data.corr()
plt.subplots(figsize=(30,10))
sns.heatmap(corr_matrix,cmap="Set3", annot = True, fmt = ".2f");
plt.title("Correlation Between Features")
plt.show()
```



In [37]:
```
threshold = 0.25
filtre = np.abs(corr_matrix["PCOS (Y/N)"]) > threshold
corr_features = corr_matrix.columns[filtre].tolist()
plt.subplots(figsize=(10,7))
sns.heatmap(data[corr_features].corr(),cmap="Set2", annot = True, fmt = ".2f")
plt.title("Correlation Between Features w Corr Theshold 0.25")
plt.show()
```



Prediction for new inputs of independent variables

In [38]:
```
new_data_1 = [[48,94.6,1,19.3,15,90,22,10.48,2,5,7,1,1,2.29,36,30,0.8323353,0.68,45
new_data_2 = [[28,44.6,152,19.3,15,78,22,10.48,2,5,0,0,1.99,1.99,36,30,0.83333,0.68
# Predict the class
new_pred_1 = lr.predict(new_data_1)
new_pred_2 = lr.predict(new_data_2)
print("Predicted classes for new data :", new_pred_1,new_pred_2)
```

```
Predicted classes for new data : [0] [0]
```

In [ ]: