In [22]:
```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from scipy import stats
#Read the data by using pandas
data = pd.read_csv(r'C:\Users\as2824\Downloads\heart_2020_cleaned.csv')
data
```

Out[22]:

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Dia
0	No	16.60	Yes	No	No	3.0	30.0	No	Female	55-59	White	
1	No	20.34	No	No	Yes	0.0	0.0	No	Female	80 or older	White	
2	No	26.58	Yes	No	No	20.0	30.0	No	Male	65-69	White	
3	No	24.21	No	No	No	0.0	0.0	No	Female	75-79	White	
4	No	23.71	No	No	No	28.0	0.0	Yes	Female	40-44	White	
...	...	...	...	...	...	...	...	...	...	...	...	
319790	Yes	27.41	Yes	No	No	7.0	0.0	Yes	Male	60-64	Hispanic	
319791	No	29.84	Yes	No	No	0.0	0.0	No	Male	35-39	Hispanic	
319792	No	24.24	No	No	No	0.0	0.0	No	Female	45-49	Hispanic	
319793	No	32.81	No	No	No	0.0	0.0	No	Female	25-29	Hispanic	
319794	No	46.56	No	No	No	0.0	0.0	No	Female	80 or older	Hispanic	

319795 rows × 18 columns

In [3]:
```
#display the first five rows in the data
print(data.head())
```

```
  HeartDisease    BMI Smoking AlcoholDrinking Stroke  PhysicalHealth  \
0           No  16.60     Yes              No     No             3.0
1           No  20.34      No              No    Yes             0.0
2           No  26.58     Yes              No     No            20.0
3           No  24.21      No              No     No             0.0
4           No  23.71      No              No     No            28.0

   MentalHealth DiffWalking     Sex  AgeCategory   Race Diabetic  \
0          30.0          No  Female        55-59  White      Yes
1           0.0          No  Female  80 or older  White       No
2          30.0          No    Male        65-69  White      Yes
3           0.0          No  Female        75-79  White       No
4           0.0         Yes  Female        40-44  White       No

  PhysicalActivity  GenHealth  SleepTime Asthma KidneyDisease SkinCancer
0              Yes  Very good        5.0    Yes            No        Yes
1              Yes  Very good        7.0     No            No         No
2              Yes       Fair        8.0    Yes            No         No
3               No       Good        6.0     No            No        Yes
4              Yes  Very good        8.0     No            No         No
```

In [4]:
```
#Display the information on no.of columns,data types, No.of rows, in our data
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 319795 entries, 0 to 319794
Data columns (total 18 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   HeartDisease      319795 non-null  object 
 1   BMI               319795 non-null  float64
 2   Smoking           319795 non-null  object 
 3   AlcoholDrinking   319795 non-null  object 
 4   Stroke            319795 non-null  object 
 5   PhysicalHealth    319795 non-null  float64
 6   MentalHealth      319795 non-null  float64
 7   DiffWalking       319795 non-null  object 
 8   Sex               319795 non-null  object 
 9   AgeCategory       319795 non-null  object 
 10  Race              319795 non-null  object 
 11  Diabetic          319795 non-null  object 
 12  PhysicalActivity  319795 non-null  object 
 13  GenHealth         319795 non-null  object 
 14  SleepTime         319795 non-null  float64
 15  Asthma            319795 non-null  object 
 16  KidneyDisease     319795 non-null  object 
 17  SkinCancer        319795 non-null  object 
dtypes: float64(4), object(14)
memory usage: 43.9+ MB
None
```

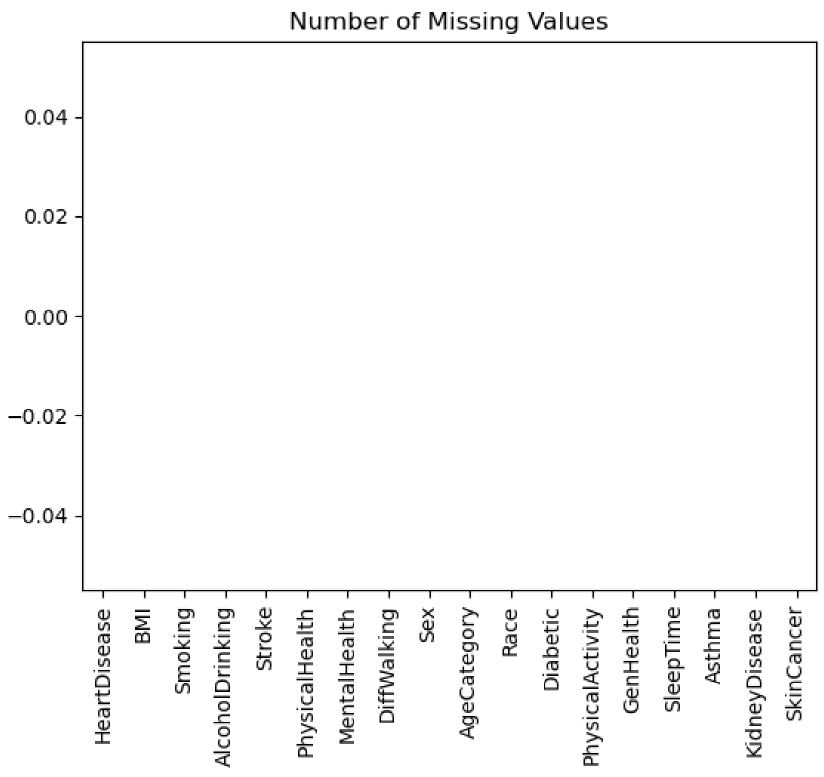
In [5]:
```
#Displayng the summary of the data in our data set
print(data.describe())
```

```
                 BMI  PhysicalHealth   MentalHealth      SleepTime
count  319795.000000    319795.00000  319795.000000  319795.000000
mean       28.325399         3.37171       3.898366       7.097075
std         6.356100         7.95085       7.955235       1.436007
min        12.020000         0.00000       0.000000       1.000000
25%        24.030000         0.00000       0.000000       6.000000
50%        27.340000         0.00000       0.000000       7.000000
75%        31.420000         2.00000       3.000000       8.000000
max        94.850000        30.00000      30.000000      24.000000
```

In [8]:
```
#Let us check if there are any missing values in our dataset
missing_values = data.isnull().sum()
print(missing_values)
missing_values.plot(kind="bar")
plt.title("Number of Missing Values")
#As there are no missing values the plot will show no data
```

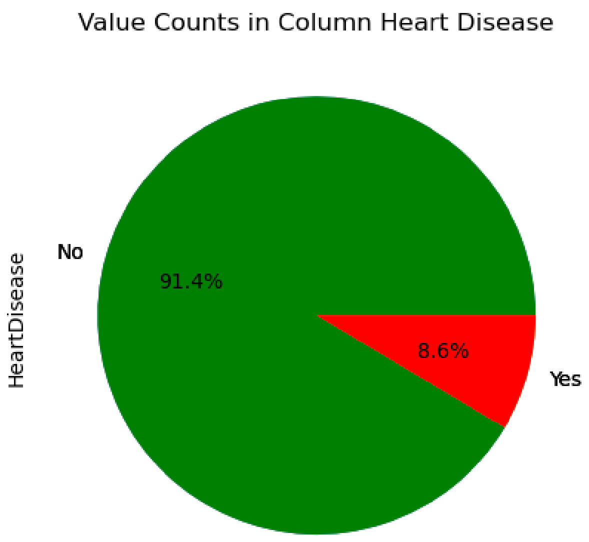
```
HeartDisease        0
BMI                 0
Smoking             0
AlcoholDrinking     0
Stroke              0
PhysicalHealth      0
MentalHealth        0
DiffWalking         0
Sex                 0
AgeCategory         0
Race                0
Diabetic            0
PhysicalActivity    0
GenHealth           0
SleepTime           0
Asthma              0
KidneyDisease       0
SkinCancer          0
dtype: int64
```

Out[8]:
```
Text(0.5, 1.0, 'Number of Missing Values')
```



In [9]:
```
# Count the frequency of each value in column HeartDisease
count = data["HeartDisease"].value_counts()
colors=["green","red"]
print(count)
# Create a pie chart to visualize the frequency of each value
count.plot(kind="pie")
plt.title("Value Counts in Column Heart Disease")
plt.pie(count, labels=count.index, colors = colors, autopct='%1.1f%%')
plt.show()
```

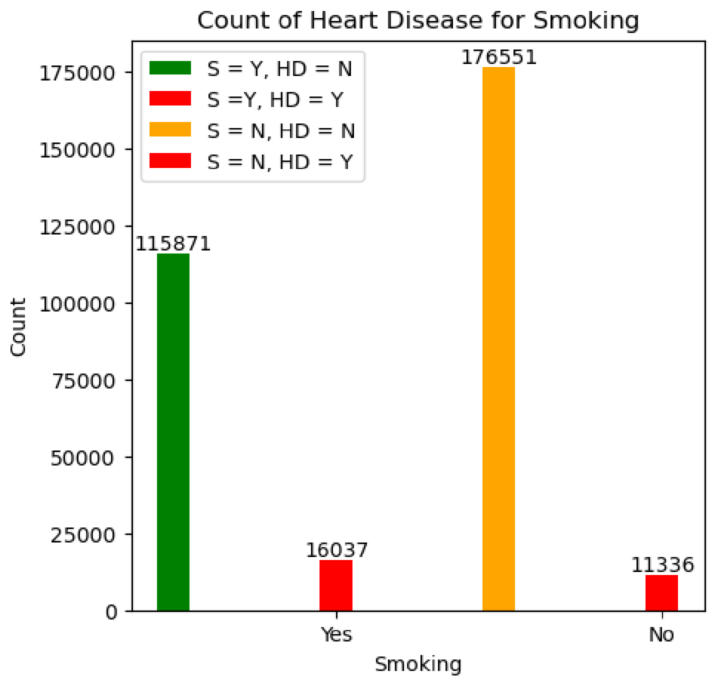
```
No     292422
Yes     27373
Name: HeartDisease, dtype: int64
```



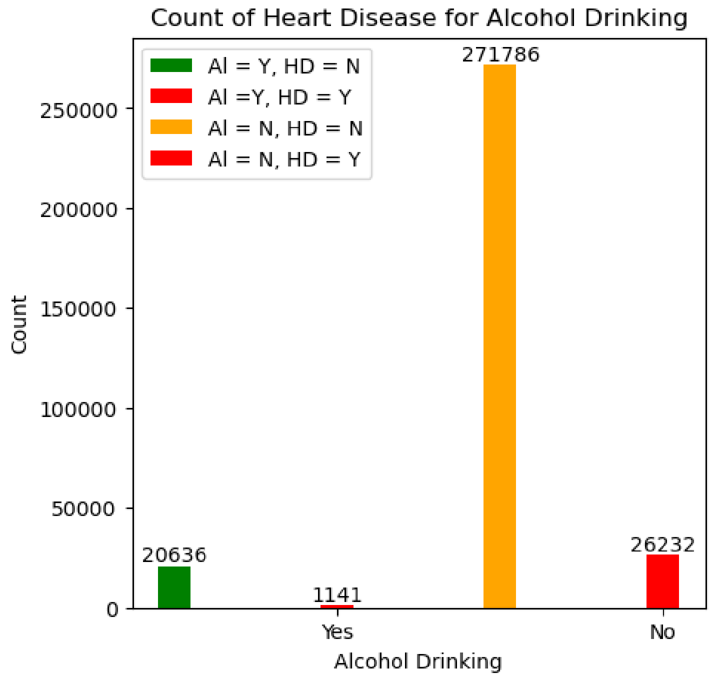
In [33]:
```
#From the above data.info() we can determine that in our data we have only 4 numerical columns and the rest are ca
Numeric_features=["BMI","PhysicalHealth","MentalHealth","SleepTime"]
for column in data.columns:
    unique_values = data[column].unique()
    total_count = data[column].nunique()
    print("The column",column,"has",total_count,"unique values","They are:",unique_values)
```

```
The column HeartDisease has 2 unique values They are: [0. 1.]
The column BMI has 3604 unique values They are: [16.6  20.34 26.58 ... 62.42 51.46 46.56]
The column Smoking has 2 unique values They are: [1. 0.]
The column AlcoholDrinking has 2 unique values They are: [0. 1.]
The column Stroke has 2 unique values They are: [0. 1.]
The column PhysicalHealth has 31 unique values They are: [ 3.  0. 20. 28.  6. 15.  5. 30.  7.  1.  2. 21.  4. 10.
14. 18.  8. 25.
 16. 29. 27. 17. 24. 12. 23. 26. 22. 19.  9. 13. 11.]
The column MentalHealth has 31 unique values They are: [30.  0.  2.  5. 15.  8.  4.  3. 10. 14. 20.  1.  7. 24.
9. 28. 16. 12.
  6. 25. 17. 18. 21. 29. 22. 13. 23. 27. 26. 11. 19.]
The column DiffWalking has 2 unique values They are: [0. 1.]
The column Sex has 2 unique values They are: ['Female' 'Male']
The column AgeCategory has 13 unique values They are: ['55-59' '80 or older' '65-69' '75-79' '40-44' '70-74' '60-
64' '50-54'
 '45-49' '18-24' '35-39' '30-34' '25-29']
The column Race has 6 unique values They are: ['White' 'Black' 'Asian' 'American Indian/Alaskan Native' 'Other'
 'Hispanic']
The column Diabetic has 4 unique values They are: [2. 0. 1. 3.]
The column PhysicalActivity has 2 unique values They are: [1. 0.]
The column GenHealth has 5 unique values They are: ['Very good' 'Fair' 'Good' 'Poor' 'Excellent']
The column SleepTime has 24 unique values They are: [ 5.  7.  8.  6. 12.  4.  9. 10. 15.  3.  2.  1. 16. 18. 14.
20. 11. 13.
 17. 24. 19. 21. 22. 23.]
The column Asthma has 2 unique values They are: [1. 0.]
The column KidneyDisease has 2 unique values They are: [0. 1.]
The column SkinCancer has 2 unique values They are: [1. 0.]
```

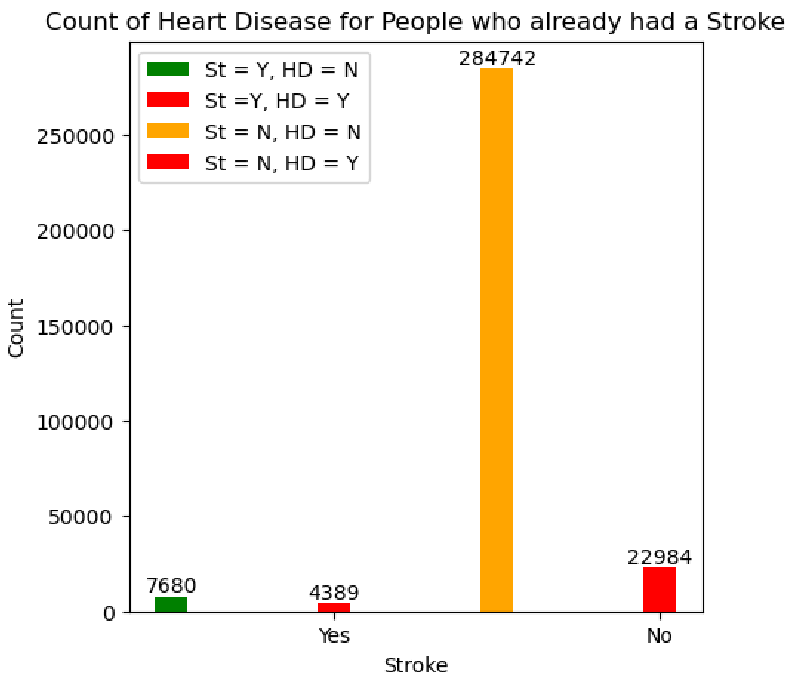
In [11]:
```
#Now we have both categorical & numeric features, we will analyse the features that are closely related to our tar
hd_smoke_yes = data[data['Smoking'] == 'Yes']['HeartDisease'].value_counts()
hd_smoke_no = data[data['Smoking'] == 'No']['HeartDisease'].value_counts()
fig, ax = plt.subplots(figsize=(5, 5))

# plot bar graph for heart disease counts for smoking yes
ax.bar(0, hd_smoke_yes['No'], color='green', label='S = Y, HD = N', width =0.2)
ax.bar(1, hd_smoke_yes['Yes'], color='red', label='S =Y, HD = Y', width=0.2)

# plot bar graph for heart disease counts for smoking no
ax.bar(2, hd_smoke_no['No'], color='orange', label = 'S = N, HD = N', width = 0.2)
ax.bar(3, hd_smoke_no['Yes'], color='red',label = 'S = N, HD = Y',width=0.2)

# add value labels on top of each bar
for i in range(4):
    ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

ax.set_title('Count of Heart Disease for Smoking')
ax.set_xticks([1,3])
ax.set_xticklabels(['Yes','No'])
ax.set_ylabel('Count')
ax.set_xlabel('Smoking')
ax.legend()
plt.show()
# The below graph shows us that following indications:
#There are 115871 samples with no Heart Disease(HD) and whose Smoking(S) = Yes
#There are 16037 samples with heart disease and whose Smoking(S) = Yes
#There are 176551 samples with no Heart Disease(HD) and whose Smoking(S) = No
#There are 11336 samples with heart disease and whose Smoking(S) = No
```



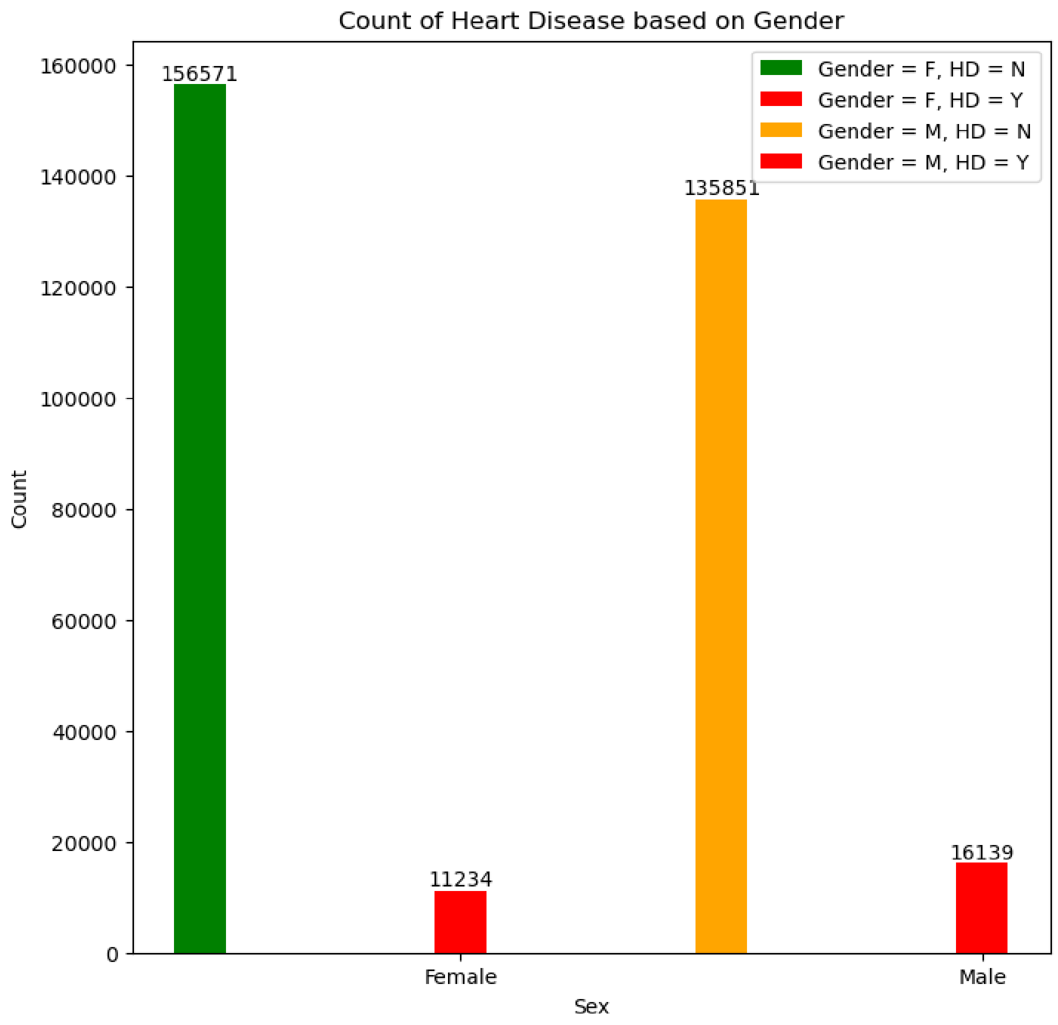
In [12]:
```
hd_al_yes = data[data['AlcoholDrinking'] == 'Yes']['HeartDisease'].value_counts()
hd_al_no = data[data['AlcoholDrinking'] == 'No']['HeartDisease'].value_counts()
fig, ax = plt.subplots(figsize=(5, 5))

# plot bar graph for heart disease counts for Alcohol Drinking yes
ax.bar(0, hd_al_yes['No'], color='green', label='Al = Y, HD = N', width =0.2)
ax.bar(1, hd_al_yes['Yes'], color='red', label='Al =Y, HD = Y', width=0.2)

# plot bar graph for heart disease counts for Alcohol Drinking no
ax.bar(2, hd_al_no['No'], color='orange', label = 'Al = N, HD = N', width = 0.2)
ax.bar(3, hd_al_no['Yes'], color='red',label = 'Al = N, HD = Y',width=0.2)

# add value labels on top of each bar
for i in range(4):
    ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

ax.set_title('Count of Heart Disease for Alcohol Drinking')
ax.set_xticks([1,3])
ax.set_xticklabels(['Yes','No'])
ax.set_ylabel('Count')
ax.set_xlabel('Alcohol Drinking')
ax.legend()
plt.show()
# The below graph shows us that following indications:
#There are 20636 samples with no Heart Disease(HD) and who Drinks Alcohol(AL)
#There are 1141 samples with heart disease and who drinks Alcohol
#There are 271786 samples with no Heart Disease(HD) and who doesn't drink Alcohol(AL)
#There are 26232 samples with heart disease and who doesn't drink Alcohol


#From the below plotted graph, we can conclude that the persons who doesn't drink Alcohol, gets the Heart Disease
#Hence, Alcohol Drinking is not considered a desired Feature for our prediction.
```



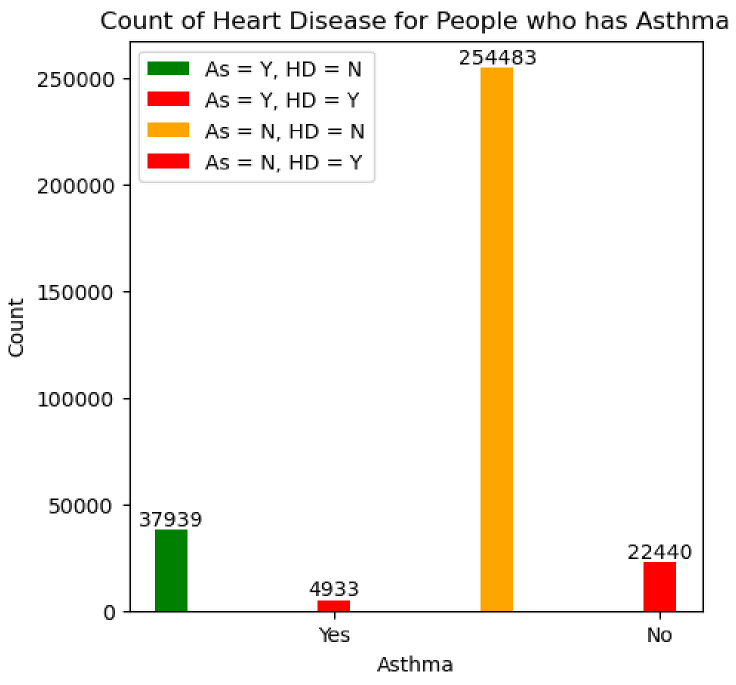
```
In [13]: hd_St_yes = data[data['Stroke'] == 'Yes']['HeartDisease'].value_counts()
         hd_St_no = data[data['Stroke'] == 'No']['HeartDisease'].value_counts()
         fig, ax = plt.subplots(figsize=(5, 5))

         # plot bar graph for heart disease counts for Stroke yes
         ax.bar(0, hd_St_yes['No'], color='green', label='St = Y, HD = N', width =0.2)
         ax.bar(1, hd_St_yes['Yes'], color='red', label='St =Y, HD = Y', width=0.2)

         # plot bar graph for heart disease counts for Stroke no
         ax.bar(2, hd_St_no['No'], color='orange', label = 'St = N, HD = N', width = 0.2)
         ax.bar(3, hd_St_no['Yes'], color='red',label = 'St = N, HD = Y',width=0.2)

         # add value labels on top of each bar
         for i in range(4):
             ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

         ax.set_title('Count of Heart Disease for People who already had a Stroke')
         ax.set_xticks([1,3])
         ax.set_xticklabels(['Yes','No'])
         ax.set_ylabel('Count')
         ax.set_xlabel('Stroke')
         ax.legend()
         plt.show()
         # The below graph shows us that following indications:
         #There are 7680 samples with no Heart Disease(HD) and who has encourtered Stroke(St)
         #There are 4389 samples with heart disease and who encourtered Stroke earlier
         #There are 284742 samples with no Heart Disease(HD) and who hasn't encourtered Stroke(St)
         #There are 22984 samples with heart disease and who hasn't encourtered Stroke(St)
```



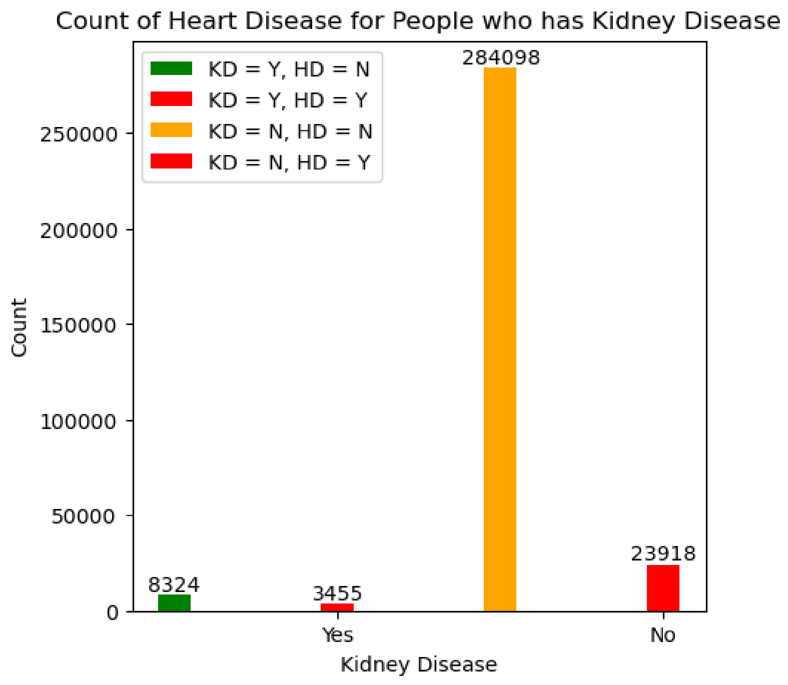
```
In [14]: hd_Gender_F = data[data['Sex'] == 'Female']['HeartDisease'].value_counts()
         hd_Gender_M = data[data['Sex'] == 'Male']['HeartDisease'].value_counts()
         fig, ax = plt.subplots(figsize=(8, 8))

         # plot bar graph for heart disease counts for Female
         ax.bar(0, hd_Gender_F['No'], color='green', label='Gender = F, HD = N', width =0.2)
         ax.bar(1, hd_Gender_F['Yes'], color='red', label='Gender = F, HD = Y', width=0.2)

         # plot bar graph for heart disease counts for Male
         ax.bar(2, hd_Gender_M['No'], color='orange', label = 'Gender = M, HD = N', width = 0.2)
         ax.bar(3, hd_Gender_M['Yes'], color='red',label = 'Gender = M, HD = Y',width=0.2)

         # add value labels on top of each bar
         for i in range(4):
             ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

         ax.set_title('Count of Heart Disease based on Gender')
         ax.set_xticks([1,3])
         ax.set_xticklabels(['Female','Male'])
         ax.set_ylabel('Count')
         ax.set_xlabel('Sex')
         ax.legend()
         plt.show()
         # The below graph shows us that following indications:
         #There are 156571 samples with no Heart Disease(HD) and Gender Female
         #There are 11234 samples with heart disease and Gender Female
         #There are 135851 samples with no Heart Disease(HD) and Gender Male
         #There are 16139 samples with heart disease and Gender Male
```



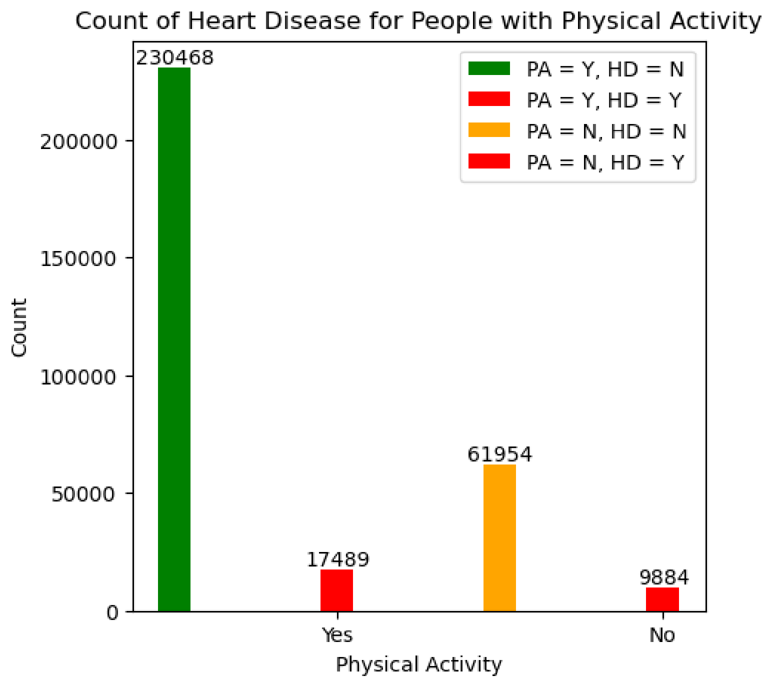
In [15]:
```
hd_As_yes = data[data['Asthma'] == 'Yes']['HeartDisease'].value_counts()
hd_As_no = data[data['Asthma'] == 'No']['HeartDisease'].value_counts()
fig, ax = plt.subplots(figsize=(5, 5))

# plot bar graph for heart disease counts for people with Asthma
ax.bar(0, hd_As_yes['No'], color='green', label='As = Y, HD = N', width =0.2)
ax.bar(1, hd_As_yes['Yes'], color='red', label='As = Y, HD = Y', width=0.2)

# plot bar graph for heart disease counts for people without Asthma
ax.bar(2, hd_As_no['No'], color='orange', label = 'As = N, HD = N', width = 0.2)
ax.bar(3, hd_As_no['Yes'], color='red',label = 'As = N, HD = Y',width=0.2)

# add value labels on top of each bar
for i in range(4):
    ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

ax.set_title('Count of Heart Disease for People who has Asthma')
ax.set_xticks([1,3])
ax.set_xticklabels(['Yes','No'])
ax.set_ylabel('Count')
ax.set_xlabel('Asthma')
ax.legend()
plt.show()
# The below graph shows us that following indications:
#There are 37939 samples with no Heart Disease(HD) and who has Asthma(AS)
#There are 4933 samples with heart disease and who has Asthma(AS)
#There are 254483 samples with no Heart Disease(HD) and who doen't have Asthma(AS)
#There are 22440 samples with heart disease and who doen't have Asthma(AS)
```



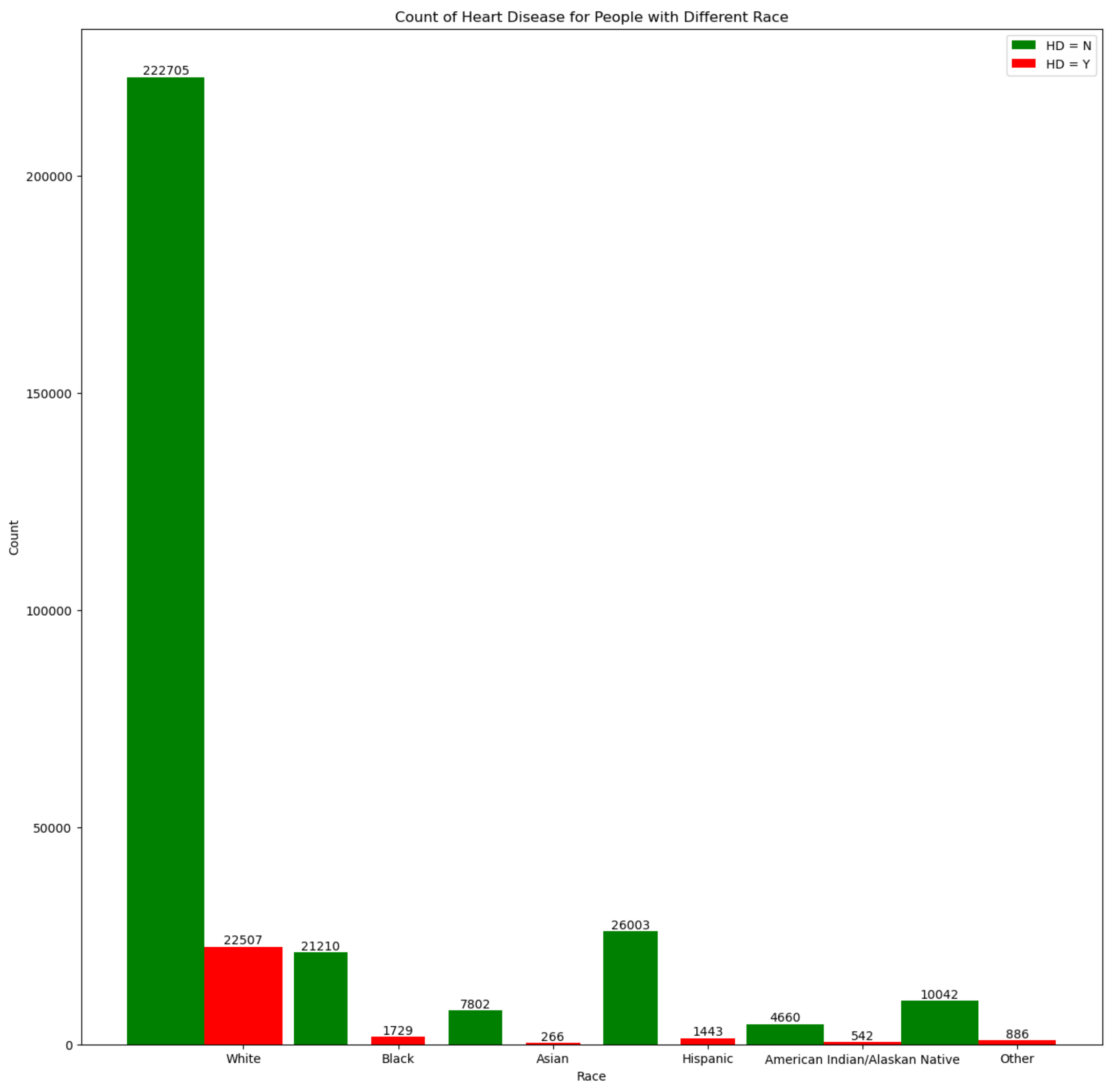
In [16]:
```python
hd_KD_yes = data[data['KidneyDisease'] == 'Yes']['HeartDisease'].value_counts()
hd_KD_no = data[data['KidneyDisease'] == 'No']['HeartDisease'].value_counts()
fig, ax = plt.subplots(figsize=(5, 5))

# plot bar graph for heart disease counts for KidneyDisease yes
ax.bar(0, hd_KD_yes['No'], color='green', label='KD = Y, HD = N', width =0.2)
ax.bar(1, hd_KD_yes['Yes'], color='red', label='KD = Y, HD = Y', width=0.2)

# plot bar graph for heart disease counts for KidneyDisease no
ax.bar(2, hd_KD_no['No'], color='orange', label = 'KD = N, HD = N', width = 0.2)
ax.bar(3, hd_KD_no['Yes'], color='red',label = 'KD = N, HD = Y',width=0.2)

# add value labels on top of each bar
for i in range(4):
    ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

ax.set_title('Count of Heart Disease for People who has Kidney Disease')
ax.set_xticks([1,3])
ax.set_xticklabels(['Yes','No'])
ax.set_ylabel('Count')
ax.set_xlabel('Kidney Disease')
ax.legend()
plt.show()
# The below graph shows us that following indications:
#There are 8324 samples with no Heart Disease(HD) and who has KidneyDisease(KD)
#There are 3455 samples with heart disease and who has KidneyDisease(KD)
#There are 284098 samples with no Heart Disease(HD) and who doen't have KidneyDisease(KD)
#There are 23918 samples with heart disease and who doen't have KidneyDisease(KD)
```



In [17]:
```python
hd_PA_yes = data[data['PhysicalActivity'] == 'Yes']['HeartDisease'].value_counts()
hd_PA_no = data[data['PhysicalActivity'] == 'No']['HeartDisease'].value_counts()
fig, ax = plt.subplots(figsize=(5, 5))

# plot bar graph for heart disease counts for Physical Activity yes
ax.bar(0, hd_PA_yes['No'], color='green', label='PA = Y, HD = N', width =0.2)
ax.bar(1, hd_PA_yes['Yes'], color='red', label='PA = Y, HD = Y', width=0.2)

# plot bar graph for heart disease counts for Physical Activity no
ax.bar(2, hd_PA_no['No'], color='orange', label = 'PA = N, HD = N', width = 0.2)
ax.bar(3, hd_PA_no['Yes'], color='red',label = 'PA = N, HD = Y',width=0.2)

# add value labels on top of each bar
for i in range(4):
    ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

ax.set_title('Count of Heart Disease for People with Physical Activity')
ax.set_xticks([1,3])
ax.set_xticklabels(['Yes','No'])
ax.set_ylabel('Count')
ax.set_xlabel('Physical Activity')
ax.legend()
plt.show()
# The below graph shows us that following indications:
#There are 230468 samples with no Heart Disease(HD) and does Physical Activity
#There are 17489 samples with heart disease and does Physical Activity
#There are 61954 samples with no Heart Disease(HD) and with no Physical Activity
#There are 9884 samples with heart disease and with no Physical Activity
```

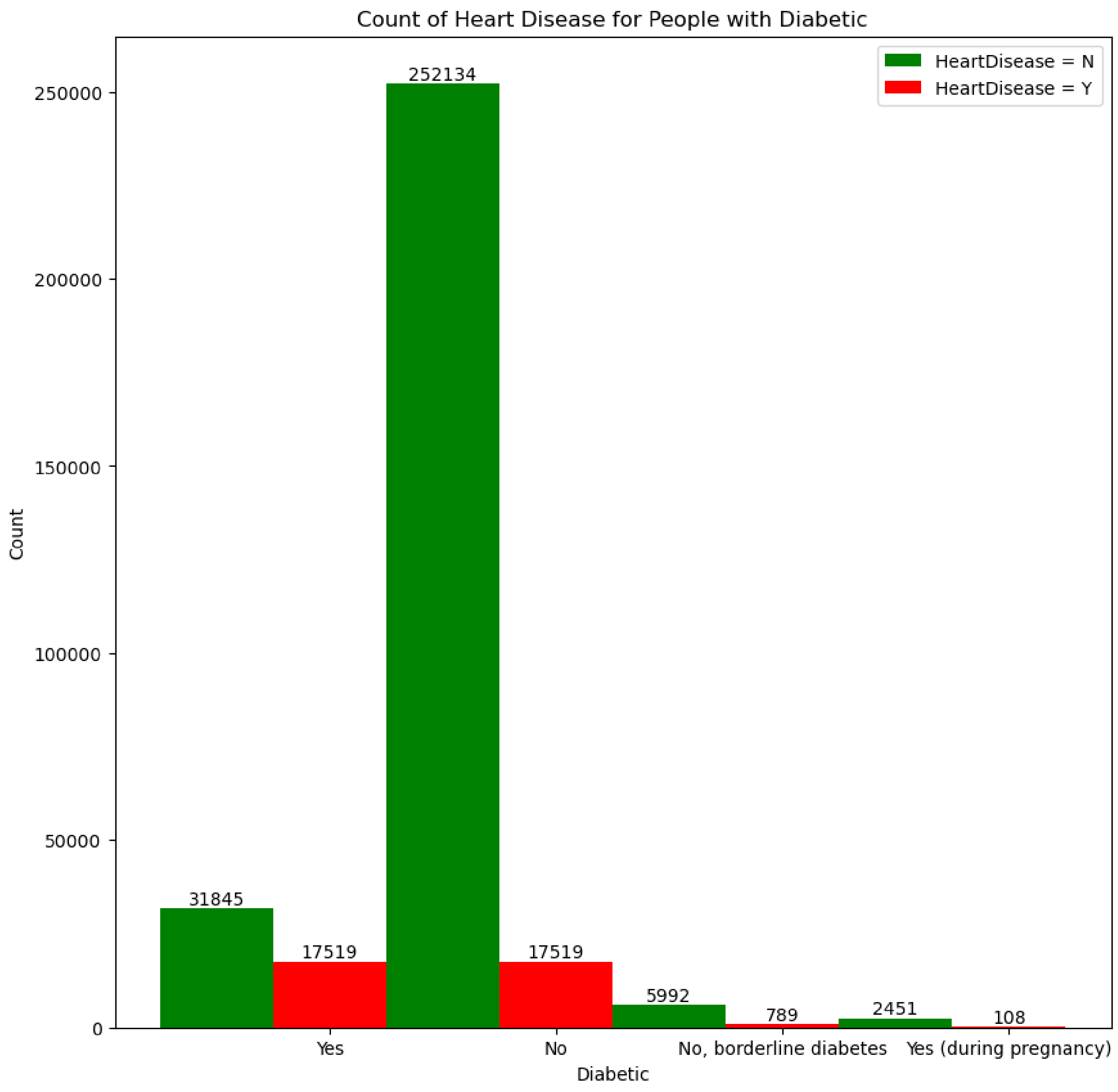


In [18]:
```python
hd_Race_White = data[data['Race'] == 'White']['HeartDisease'].value_counts()
hd_Race_Black = data[data['Race'] == 'Black']['HeartDisease'].value_counts()
hd_Race_Asian = data[data['Race'] == 'Asian']['HeartDisease'].value_counts()
hd_Race_Hispanic = data[data['Race'] == 'Hispanic']['HeartDisease'].value_counts()
hd_Race_AI_AN = data[data['Race'] == 'American Indian/Alaskan Native']['HeartDisease'].value_counts()
hd_Race_other = data[data['Race'] == 'Other']['HeartDisease'].value_counts()


fig, ax = plt.subplots(figsize=(15, 15))

# plot bar graph for heart disease counts for race White
ax.bar(0, hd_Race_White['No'], color='green', label='HD = N', width =1.0)
ax.bar(1, hd_Race_White['Yes'], color='red', label='HD = Y', width=1.0)
# plot bar graph for heart disease counts for race Black
ax.bar(2, hd_Race_Black['No'], color='green', width = 0.7)
ax.bar(3, hd_Race_Black['Yes'], color='red', width=0.7)
# plot bar graph for heart disease counts for race Asian
ax.bar(4, hd_Race_Asian['No'], color='green',width = 0.7)
ax.bar(5, hd_Race_Asian['Yes'], color='red', width=0.7)
# plot bar graph for heart disease counts for race Hispanic
ax.bar(6, hd_Race_Hispanic['No'], color='green',width = 0.7)
ax.bar(7, hd_Race_Hispanic['Yes'], color='red', width = 0.7)
# plot bar graph for heart disease counts for race American Indian/Alaskan Native
ax.bar(8, hd_Race_AI_AN['No'], color='green', width =1.0)
ax.bar(9, hd_Race_AI_AN['Yes'], color='red', width=1.0)
# plot bar graph for heart disease counts for race others
ax.bar(10, hd_Race_other['No'], color='green',width = 1.0)
ax.bar(11, hd_Race_other['Yes'], color='red',width=1.0)

# add value labels on top of each bar
for i in range(12):
    ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

ax.set_title('Count of Heart Disease for People with Different Race')
ax.set_xticks([1,3,5,7,9,11])
ax.set_xticklabels(['White','Black','Asian','Hispanic','American Indian/Alaskan Native', 'Other'])
ax.set_ylabel('Count')
ax.set_xlabel('Race')
ax.legend()
plt.show()
```

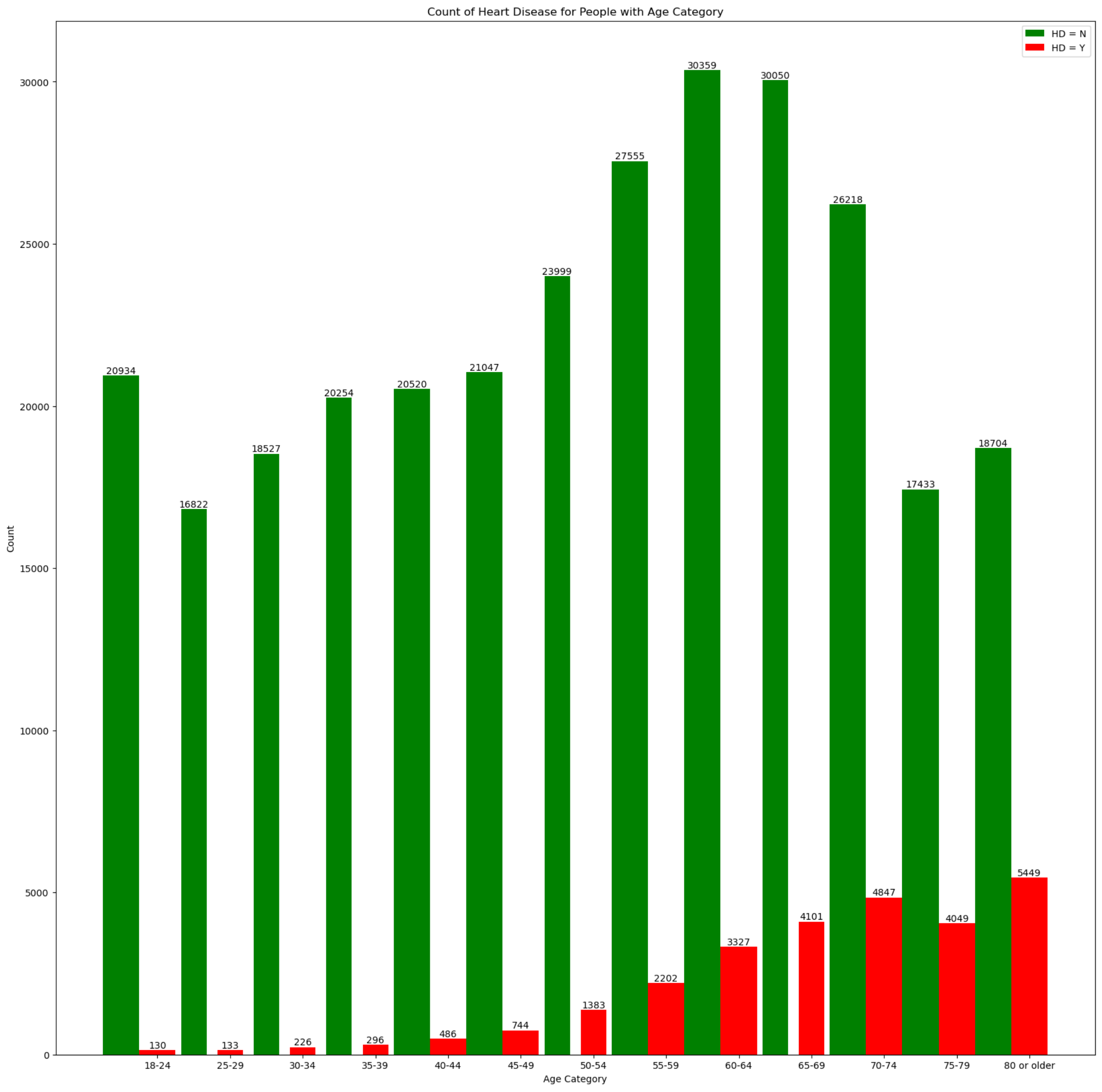

Count of Heart Disease for People with Different Race
HD = N
HD = Y
Count
Race
White
Black
Asian
Hispanic
American Indian/Alaskan Native
Other
222705
22507
21210
1729
7802
266
26003
1443
4660
542
10042
886
0
50000
100000
150000
200000

In [19]:

```python
hd_Diabetic_Yes= data[data['Diabetic'] == 'Yes']['HeartDisease'].value_counts()
hd_Diabetic_No = data[data['Diabetic'] == 'No']['HeartDisease'].value_counts()
hd_Diabetic_bd = data[data['Diabetic'] == 'No, borderline diabetes']['HeartDisease'].value_counts()
hd_Diabetic_yes_dp = data[data['Diabetic'] == 'Yes (during pregnancy)']['HeartDisease'].value_counts()


fig, ax = plt.subplots(figsize=(10, 10))

# plot bar graph for heart disease counts for race White
ax.bar(0, hd_Diabetic_Yes['No'], color='green', label='HeartDisease = N', width =1.0)
ax.bar(1, hd_Diabetic_No['Yes'], color='red', label='HeartDisease = Y', width=1.0)
# plot bar graph for heart disease counts for race Black
ax.bar(2, hd_Diabetic_No['No'], color='green', width = 1.0)
ax.bar(3, hd_Diabetic_No['Yes'], color='red', width=1.0)
# plot bar graph for heart disease counts for race Asian
ax.bar(4, hd_Diabetic_bd['No'], color='green',width = 1.0)
ax.bar(5,hd_Diabetic_bd ['Yes'], color='red', width=1.0)
# plot bar graph for heart disease counts for race Hispanic
ax.bar(6, hd_Diabetic_yes_dp['No'], color='green',width = 1.0)
ax.bar(7, hd_Diabetic_yes_dp['Yes'], color='red', width = 1.0)

for i in range(8):
    ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

ax.set_title('Count of Heart Disease for People with Diabetic')
ax.set_xticks([1,3,5,7])
ax.set_xticklabels(['Yes','No','No, borderline diabetes','Yes (during pregnancy)'])
ax.set_ylabel('Count')
ax.set_xlabel('Diabetic')
ax.legend()
plt.show()
```


Count of Heart Disease for People with Diabetic
HeartDisease = N
HeartDisease = Y
252134
31845
17519
17519
5992
789
2451
108
0
50000
100000
150000
200000
250000
Count
Yes
No
No, borderline diabetes
Yes (during pregnancy)
Diabetic

In [23]:
```
hd_Age_18_24 = data[data['AgeCategory'] == '18-24']['HeartDisease'].value_counts()
hd_Age_25_29 = data[data['AgeCategory'] == '25-29']['HeartDisease'].value_counts()
hd_Age_30_34 = data[data['AgeCategory'] == '30-34']['HeartDisease'].value_counts()
hd_Age_35_39 = data[data['AgeCategory'] == '35-39']['HeartDisease'].value_counts()
hd_Age_40_44 = data[data['AgeCategory'] == '40-44']['HeartDisease'].value_counts()
hd_Age_45_49 = data[data['AgeCategory'] == '45-49']['HeartDisease'].value_counts()
hd_Age_50_54 = data[data['AgeCategory'] == '50-54']['HeartDisease'].value_counts()
hd_Age_55_59 = data[data['AgeCategory'] == '55-59']['HeartDisease'].value_counts()
hd_Age_60_64 = data[data['AgeCategory'] == '60-64']['HeartDisease'].value_counts()
hd_Age_65_69 = data[data['AgeCategory'] == '65-69']['HeartDisease'].value_counts()
hd_Age_70_74 = data[data['AgeCategory'] == '70-74']['HeartDisease'].value_counts()
hd_Age_75_79 = data[data['AgeCategory'] == '75-79']['HeartDisease'].value_counts()
hd_Age_80_older = data[data['AgeCategory'] == '80 or older']['HeartDisease'].value_counts()


fig, ax = plt.subplots(figsize=(20, 20))

# plot bar graph for heart disease counts for race White
ax.bar(0, hd_Age_18_24['No'], color='green', label='HD = N', width =1.0)
ax.bar(1, hd_Age_18_24['Yes'], color='red', label='HD = Y', width=1.0)
# plot bar graph for heart disease counts for race Black
ax.bar(2, hd_Age_25_29['No'], color='green', width = 0.7)
ax.bar(3, hd_Age_25_29['Yes'], color='red', width=0.7)
# plot bar graph for heart disease counts for race Asian
ax.bar(4, hd_Age_30_34['No'], color='green',width = 0.7)
ax.bar(5, hd_Age_30_34['Yes'], color='red', width=0.7)
# plot bar graph for heart disease counts for race Hispanic
ax.bar(6, hd_Age_35_39['No'], color='green',width = 0.7)
ax.bar(7, hd_Age_35_39['Yes'], color='red', width = 0.7)
# plot bar graph for heart disease counts for race American Indian/Alaskan Native
ax.bar(8, hd_Age_40_44['No'], color='green', width =1.0)
ax.bar(9, hd_Age_40_44['Yes'], color='red', width=1.0)
# plot bar graph for heart disease counts for race others
ax.bar(10, hd_Age_45_49['No'], color='green',width = 1.0)
ax.bar(11, hd_Age_45_49['Yes'], color='red',width=1.0)
# plot bar graph for heart disease counts for race Hispanic
ax.bar(12, hd_Age_50_54['No'], color='green',width = 0.7)
ax.bar(13, hd_Age_50_54['Yes'], color='red', width = 0.7)
# plot bar graph for heart disease counts for race American Indian/Alaskan Native
ax.bar(14, hd_Age_55_59['No'], color='green', width =1.0)
ax.bar(15, hd_Age_55_59['Yes'], color='red', width=1.0)
# plot bar graph for heart disease counts for race others
ax.bar(16, hd_Age_60_64['No'], color='green',width = 1.0)
ax.bar(17, hd_Age_60_64['Yes'], color='red',width=1.0)
# plot bar graph for heart disease counts for race Hispanic
ax.bar(18, hd_Age_65_69['No'], color='green',width = 0.7)
ax.bar(19, hd_Age_65_69['Yes'], color='red', width = 0.7)
# plot bar graph for heart disease counts for race American Indian/Alaskan Native
ax.bar(20, hd_Age_70_74['No'], color='green', width =1.0)
ax.bar(21, hd_Age_70_74['Yes'], color='red', width=1.0)
# plot bar graph for heart disease counts for race others
ax.bar(22, hd_Age_75_79['No'], color='green',width = 1.0)
ax.bar(23, hd_Age_75_79['Yes'], color='red',width=1.0)
# plot bar graph for heart disease counts for race others
ax.bar(24, hd_Age_80_older['No'], color='green',width = 1.0)
ax.bar(25, hd_Age_80_older['Yes'], color='red',width=1.0)


# add value labels on top of each bar
for i in range(26):
    ax.text(i, ax.patches[i].get_height(), ax.patches[i].get_height(), ha='center', va='bottom')

ax.set_title('Count of Heart Disease for People with Age Category')
ax.set_xticks([1,3,5,7,9,11,13,15,17,19,21,23,25])
ax.set_xticklabels(['18-24','25-29','30-34','35-39','40-44','45-49','50-54','55-59','60-64','65-69','70-74','75-79
ax.set_ylabel('Count')
ax.set_xlabel('Age Category')
ax.legend()
plt.show()
```

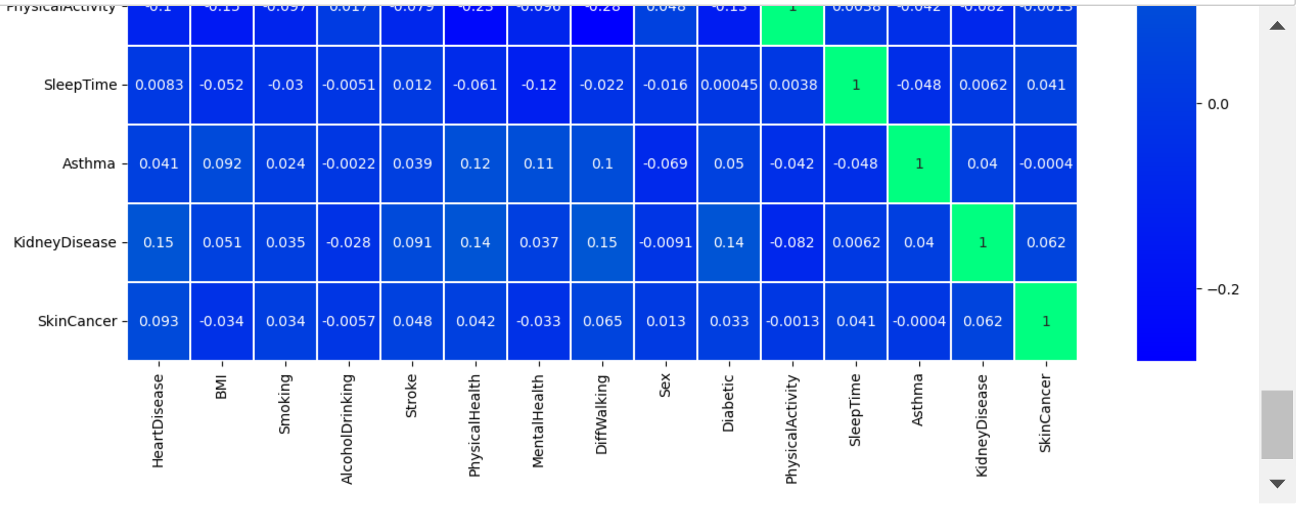

Count of Heart Disease for People with Age Category
HD = N
HD = Y
Count
Age Category
0
5000
10000
15000
20000
25000
30000
18-24
25-29
30-34
35-39
40-44
45-49
50-54
55-59
60-64
65-69
70-74
75-79
80 or older
20934
130
16822
133
18527
226
20254
296
20520
486
21047
744
23999
1383
27555
2202
30359
3327
30050
4101
26218
4847
17433
4049
18704
5449

In [24]:
```python
from sklearn.preprocessing import OrdinalEncoder
data.fillna("missing", inplace=True)

import pandas as pd
Categorical_features=['HeartDisease', 'Smoking', 'AlcoholDrinking', 'Stroke', 'Sex', 'DiffWalking', 'Diabetic','Ph
# create an instance of the encoder
encoding = OrdinalEncoder()

# fit the encoder to the data
encoding_categorical_features = encoding.fit_transform(data[Categorical_features])
# transform the categorical features
data[Categorical_features] = encoding_categorical_features

data.head(10)
```

Out[24]:

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	Phys
0	0.0	16.60	1.0	0.0	0.0	3.0	30.0	0.0	0.0	55-59	White	2.0	
1	0.0	20.34	0.0	0.0	1.0	0.0	0.0	0.0	0.0	80 or older	White	0.0	
2	0.0	26.58	1.0	0.0	0.0	20.0	30.0	0.0	1.0	65-69	White	2.0	
3	0.0	24.21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	75-79	White	0.0	
4	0.0	23.71	0.0	0.0	0.0	28.0	0.0	1.0	0.0	40-44	White	0.0	
5	1.0	28.87	1.0	0.0	0.0	6.0	0.0	1.0	0.0	75-79	Black	0.0	
6	0.0	21.63	0.0	0.0	0.0	15.0	0.0	0.0	0.0	70-74	White	0.0	
7	0.0	31.64	1.0	0.0	0.0	5.0	0.0	1.0	0.0	80 or older	White	2.0	
8	0.0	26.45	0.0	0.0	0.0	0.0	0.0	0.0	0.0	80 or older	White	1.0	
9	0.0	40.69	0.0	0.0	0.0	0.0	0.0	1.0	1.0	65-69	White	0.0	

In [70]:
```python
df = data.copy()

# calculate correlation matrix
correlation_mat = df.corr()

# print correlation table
print(correlation_mat)
plt.figure(figsize=(15,15))
sns.heatmap(df.corr(),linewidth=.05,annot = True, cmap="winter")
plt.show()
plt.savefig('correlationfigure')
```



In [90]:
```
#From the above correlation matrix, we can see that the following attributes are correlated highly with the attribu
Highly_correlated = ["HeartDisease","Smoking","Stroke","DiffWalking","Sex","Diabetic","KidneyDisease","SkinCancer"
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Select only the relevant attributes
df = df[Highly_correlated]

# Define a mapping for converting user input to numerical values
mapp = {"yes": 1, "no": 0, "male": 1, "female": 0}

# Split the data into training and testing sets
X = df.drop("HeartDisease", axis=1)
y = df["HeartDisease"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5)

# Fit a Gaussian Naive Bayes model on the training set
gnb = GaussianNB()
gnb.fit(X_train, y_train)

# Take input from the user
smoke = input("Does the patient smoke?(yes/no): ").lower()
strke = input("Does the patient previously have heart attack?(yes/no): ").lower()
difwk = input("Do the patient have difficulty in walking?(yes/no): ").lower()
dbtc = input("Is the patient diabetic?(yes/no): ").lower()
sex = input("Gender of the patient?(male/female): ").lower()
kd = input("Does the patient have kidney disease(yes/no): ").lower()
sc = input("Did the patient have skin cancer?(yes/no): ").lower()

# Convert the user inputs to numerical values using the mapping
input_data = pd.DataFrame({"Smoking": [mapping[smoke]], "Stroke": [mapping[strke]], "DiffWalking": [mapping[difwk]

# Predict the heart disease status
prediction_output = gnb.predict(input_data)

# Return the prediction to the user
if prediction_output[0] == 1:
    print("The patient is at a high risk for heart disease.")
else:
    print("The patient is at a low risk for heart disease.")

gnb.fit(X_train, y_train)
# Make predictions on the testing set
y_pred = gnb.predict(X_test)
# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", np.round(accuracy*100,2),"%")
from warnings import simplefilter
# ignore all future warnings so that we can see the result directly without warnings
simplefilter(action='ignore', category=FutureWarning)
```

```
Does the patient smoke?(yes/no): yes
Does the patient previously have heart attack?(yes/no): no
Do the patient have difficulty in walking?(yes/no): no
Is the patient diabetic?(yes/no): no
Gender of the patient?(male/female): male
Does the patient have kidney disease(yes/no): no
Did the patient have skin cancer?(yes/no): no
The patient is at a low risk for heart disease.
Accuracy: 87.09 %
```

In [ ]: