# Data Analysis Of Framingham Heart Study

**Author: Sohel Japanwala**

**Email: sohel.japanwala@gmail.com**

**LinkedIn: https://www.linkedin.com/in/soheljapanwala/ (https://www.linkedin.com/in/soheljapanwala/)**

# 1. Introduction

We'll describe the Framingham Heart Study, one of the most important epidemiological studies ever conducted, and the underlying analytics that led to our current understanding of cardiovascular disease.

In the late 1940s, the US government set out to better understand cardiovascular disease. The plan was to track a large cohort of initially healthy patients over their lifetimes. A city was chosen, the city of Framingham, Massachusetts, to be the site for the study. Framingham has an appropriate size. It's not too large, it's not too small. It has a stable population that doesn't move too much. And the doctors and residents were quite cooperative. So in 1948, the Framingham Heart Study started.

The study included 5,209 patients, aged 30 to 59. Patients were given a questionnaire and an examination every two years. During this examination, their physical characteristics were recorded, their behavioral characteristics, as well as test results.

## 2. Problem Statement

- We determine category wise numbers
- We plot graphs for various segregation
- We derive various insights from the study affect various health parameters

## 3. Data Pre Processing

## 3.1 Import Packages



In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

## 3.2 Import Data

In [2]:
```
framinghamDf=pd.read_csv("framingham.csv")
```

## 3.3 Data Preview

In [3]:
```
framinghamDf.head()
```

Out[3]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | di |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 39 | 4.0 | 0 | 0.0 | 0.0 | 0 | 0 | |
| **1** | 0 | 46 | 2.0 | 0 | 0.0 | 0.0 | 0 | 0 | |
| **2** | 1 | 48 | 1.0 | 1 | 20.0 | 0.0 | 0 | 0 | |
| **3** | 0 | 61 | 3.0 | 1 | 30.0 | 0.0 | 0 | 1 | |
| **4** | 0 | 46 | 3.0 | 1 | 23.0 | 0.0 | 0 | 0 | |

## 3.4 Data Shape

In [4]:
```
framinghamDf.shape
```

Out[4]: (4240, 16)

## 3.5 Data Description

In [5]:
```
framinghamDf.describe()
```

Out[5]:

| | male | | | | | | |
|---|---|---|---|---|---|---|---|
| **count** | 4240.000000 | 4240.000000 | 4135.000000 | 4240.000000 | 4211.000000 | 4187.000000 | 424 |
| **mean** | 0.429245 | 49.580189 | 1.979444 | 0.494104 | 9.005937 | 0.029615 | |
| **std** | 0.495027 | 8.572942 | 1.019791 | 0.500024 | 11.922462 | 0.169544 | |
| **min** | 0.000000 | 32.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **25%** | 0.000000 | 42.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **50%** | 0.000000 | 49.000000 | 2.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **75%** | 1.000000 | 56.000000 | 3.000000 | 1.000000 | 20.000000 | 0.000000 | |
| **max** | 1.000000 | 70.000000 | 4.000000 | 1.000000 | 70.000000 | 1.000000 | |



## 3.6 Data Description

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4240 entries, 0 to 4239
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4240 non-null   int64  
 1   age              4240 non-null   int64  
 2   education        4135 non-null   float64
 3   currentSmoker    4240 non-null   int64  
 4   cigsPerDay       4211 non-null   float64
 5   BPMeds           4187 non-null   float64
 6   prevalentStroke  4240 non-null   int64  
 7   prevalentHyp     4240 non-null   int64  
 8   diabetes         4240 non-null   int64  
 9   totChol          4190 non-null   float64
 10  sysBP            4240 non-null   float64
 11  diaBP            4240 non-null   float64
 12  BMI              4221 non-null   float64
 13  heartRate        4239 non-null   float64
 14  glucose          3852 non-null   float64
 15  TenYearCHD       4240 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 530.1 KB
```

## Observations On Data Description



- **education, cigsPerDay, BPMeds, totChol, BMI, heartRate, TenYearCHD** columns have **missing data**

# 3.7 Data Cleaning

### 3.7.1 Replacing Missing Data

In [7]:
```
framinghamDf.fillna("NaN")
```

Out[7]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 39 | 4 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 46 | 2 | 0 | 0 | 0 | 0 | 0 |
| **2** | 1 | 48 | 1 | 1 | 20 | 0 | 0 | 0 |
| **3** | 0 | 61 | 3 | 1 | 30 | 0 | 0 | 1 |
| **4** | 0 | 46 | 3 | 1 | 23 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **4235** | 0 | 48 | 2 | 1 | 20 | NaN | 0 | 0 |
| **4236** | 0 | 44 | 1 | 1 | 15 | 0 | 0 | 0 |
| **4237** | 0 | 52 | 2 | 0 | 0 | 0 | 0 | 0 |
| **4238** | 1 | 40 | 3 | 0 | 0 | 0 | 0 | 1 |
| **4239** | 0 | 39 | 3 | 1 | 30 | 0 | 0 | 0 |

4240 rows × 16 columns

In [8]:
```
framinghamDf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4240 entries, 0 to 4239
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   male             4240 non-null   int64
 1   age              4240 non-null   int64
 2   education        4135 non-null   float64
 3   currentSmoker    4240 non-null   int64
 4   cigsPerDay       4211 non-null   float64
 5   BPMeds           4187 non-null   float64
 6   prevalentStroke  4240 non-null   int64
 7   prevalentHyp     4240 non-null   int64
 8   diabetes         4240 non-null   int64
 9   totChol          4190 non-null   float64
 10  sysBP            4240 non-null   float64
 11  diaBP            4240 non-null   float64
 12  BMI              4221 non-null   float64
 13  heartRate        4239 non-null   float64
 14  glucose          3852 non-null   float64
 15  TenYearCHD       4240 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 530.1 KB
```



## 3.7.2 Adding Serial Number Column

In [9]:
```
framinghamDf["Serial"]=framinghamDf.index+1
framinghamDf.head()
```

Out[9]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | di |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 39 | 4.0 | 0 | 0.0 | 0.0 | 0 | 0 | |
| **1** | 0 | 46 | 2.0 | 0 | 0.0 | 0.0 | 0 | 0 | |
| **2** | 1 | 48 | 1.0 | 1 | 20.0 | 0.0 | 0 | 0 | |
| **3** | 0 | 61 | 3.0 | 1 | 30.0 | 0.0 | 0 | 1 | |
| **4** | 0 | 46 | 3.0 | 1 | 23.0 | 0.0 | 0 | 0 | |

## Observations:

- Missing data has been replaced
- Cloumns as required have been added

# 4. Data Analysis



### 4.1 Analyzing Gender Wise Da

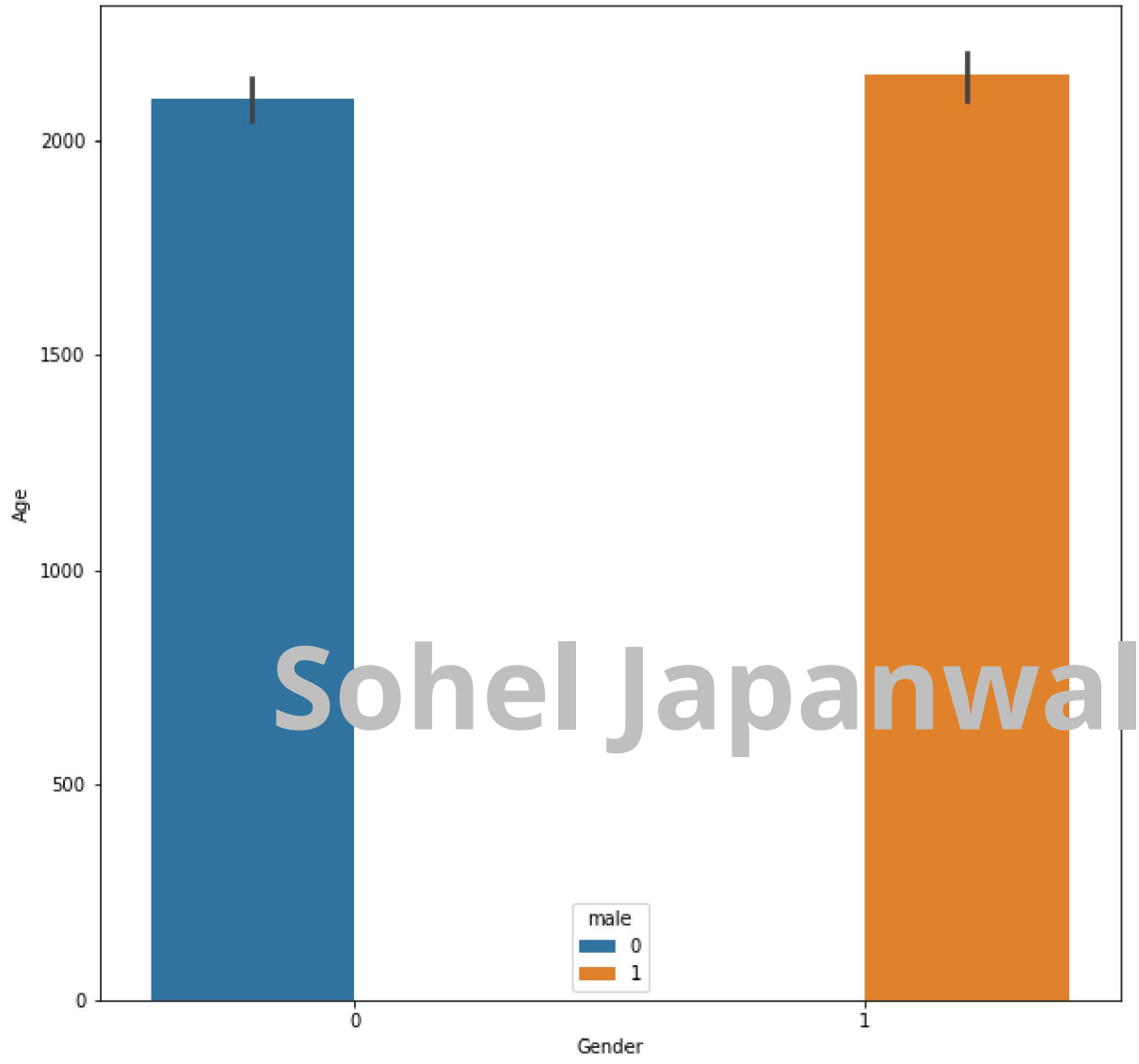
In [10]:
```
pd.DataFrame(framinghamDf.groupby("male")["Serial"].nunique())
```

Out[10]:

| male | Serial |
|---|---|
| **0** | 2420 |
| **1** | 1820 |

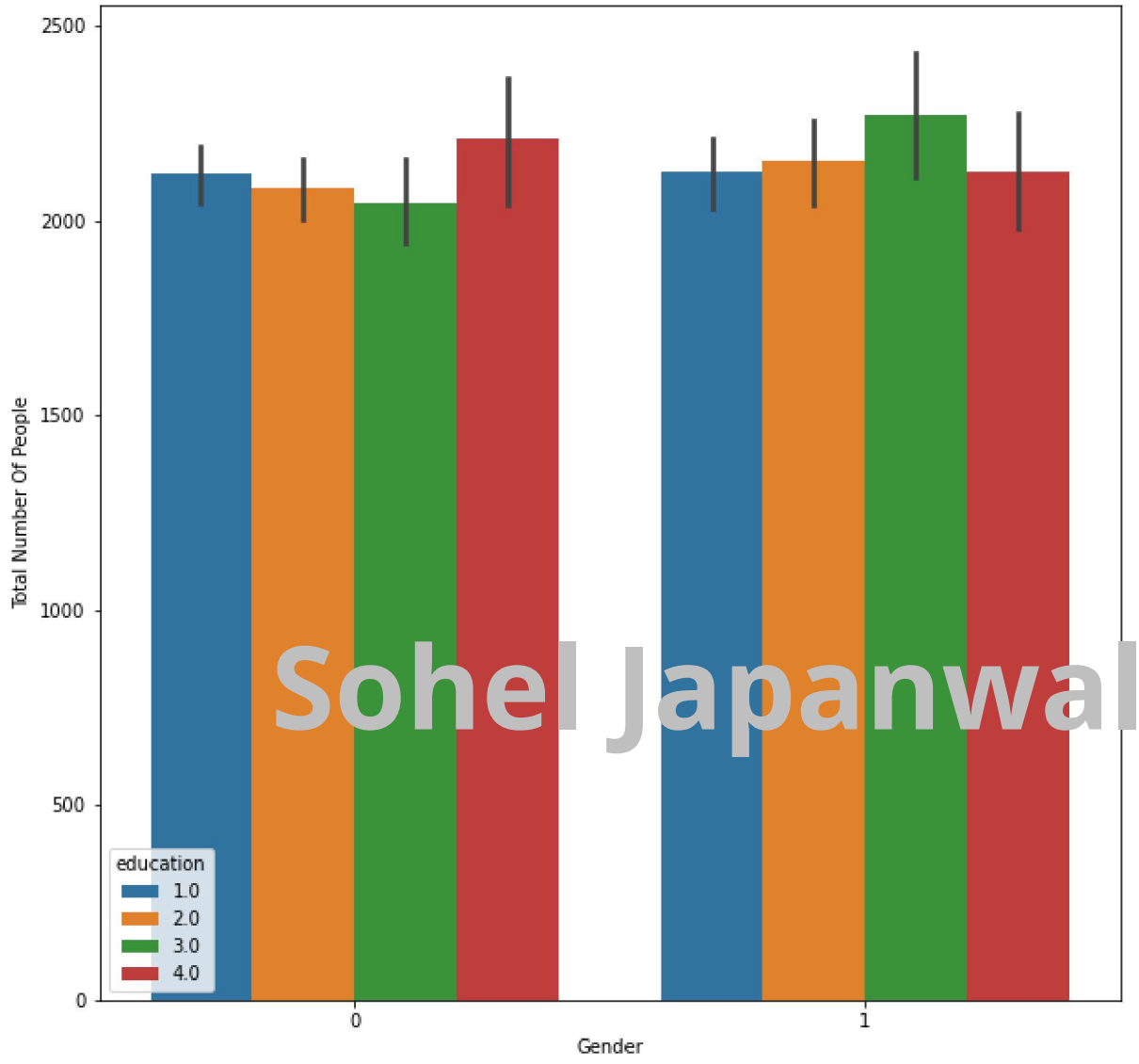
In [11]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["Serial"], hue=framinghamDf[
"male"])
plt.ylabel("Age")
plt.xlabel("Gender")
```

Out[11]: Text(0.5, 0, 'Gender')



In [12]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["Serial"], hue=framinghamDf[
"education"])
plt.ylabel("Total Number Of People")
plt.xlabel("Gender")
```

Out[12]: Text(0.5, 0, 'Gender')





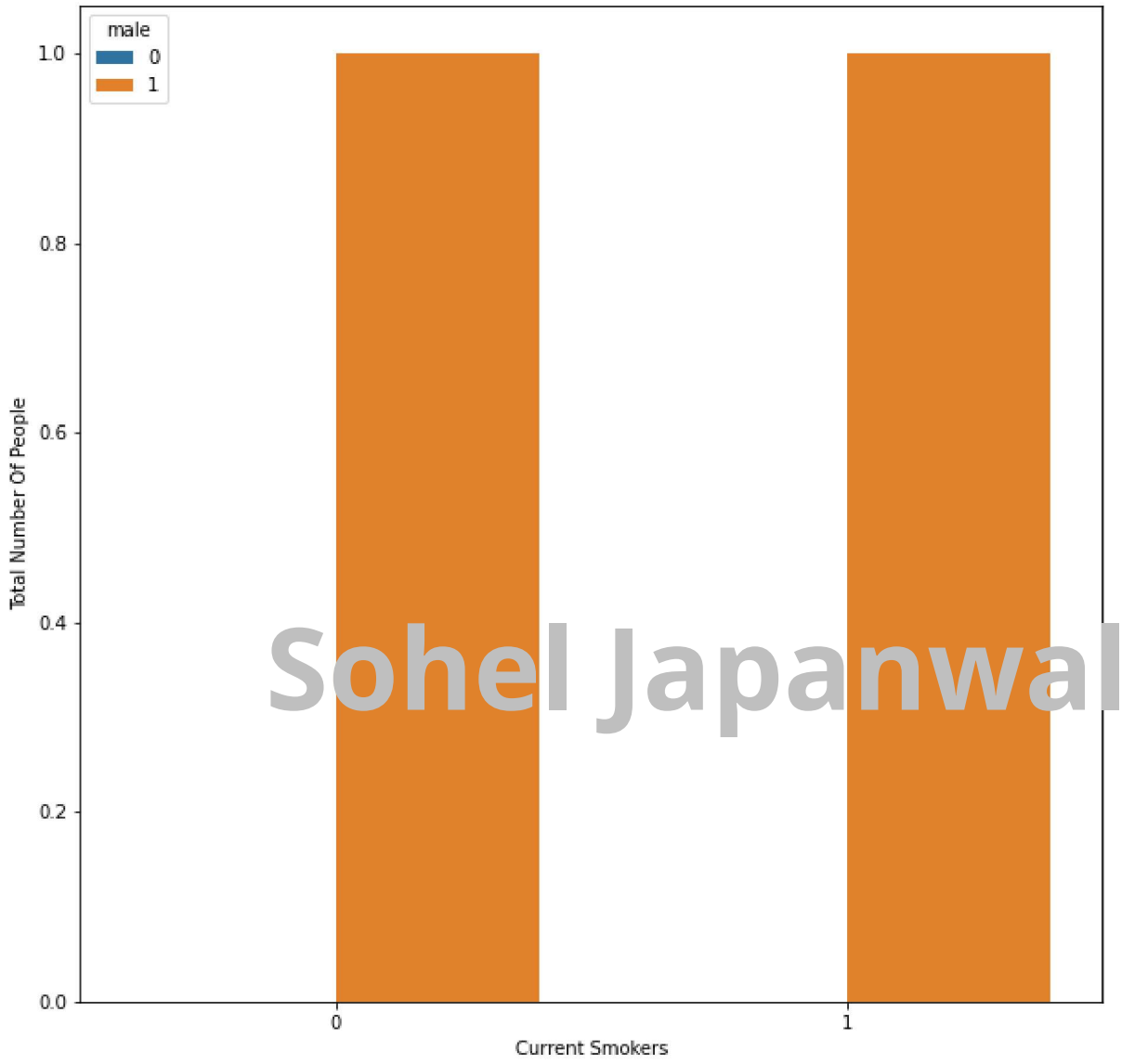

In [13]:
```
pd.DataFrame(framinghamDf.groupby("education")["Serial"].nunique())
```

Out[13]:

| education | Serial |
|---|---|
| **1.0** | 1720 |
| **2.0** | 1253 |
| **3.0** | 689 |
| **4.0** | 473 |

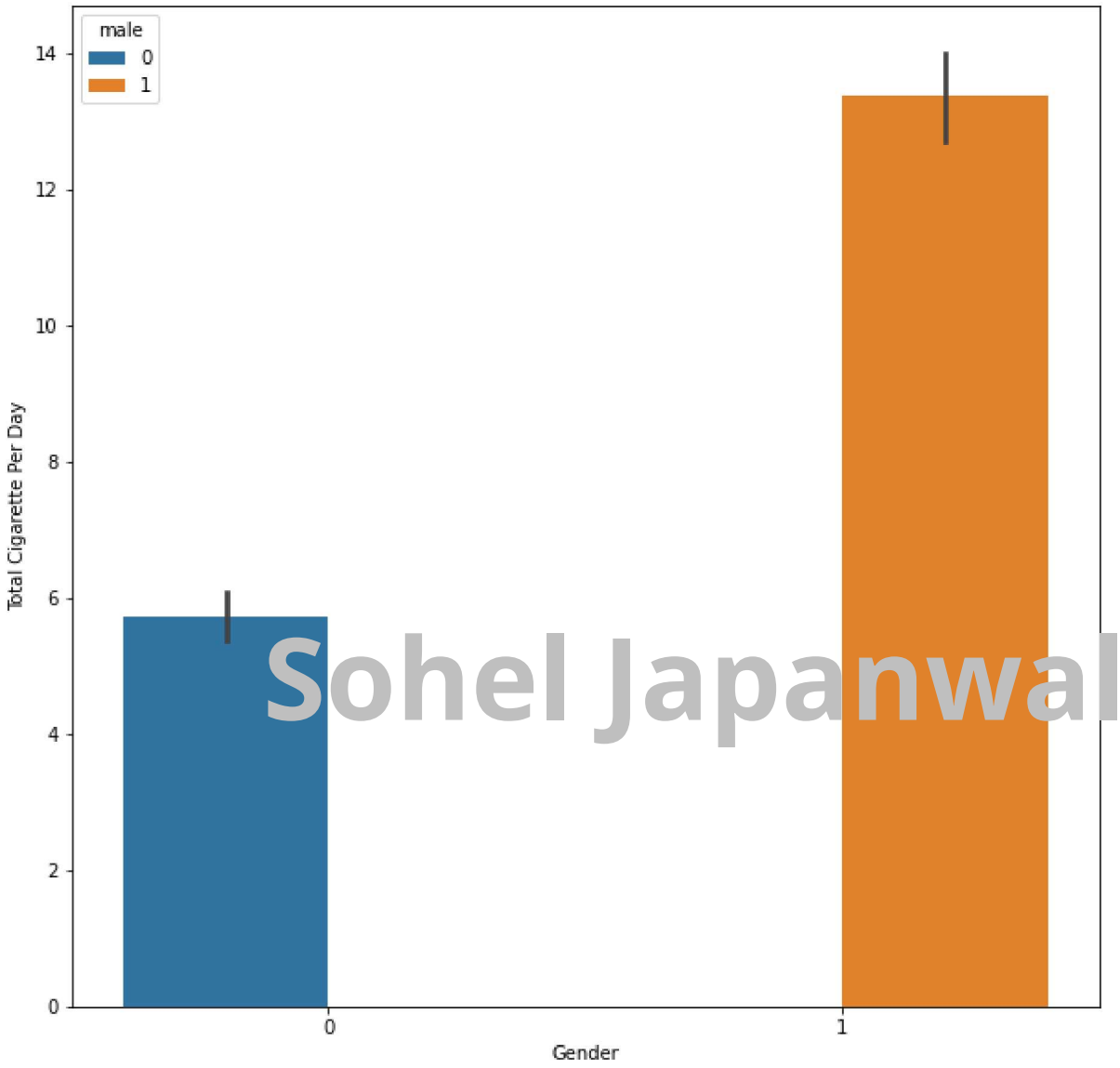
In [14]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["currentSmoker"],y=framinghamDf["male"], hue=framin
ghamDf["male"])
plt.ylabel("Total Number Of People")
plt.xlabel("Current Smokers")
```

Out[14]: Text(0.5, 0, 'Current Smokers')



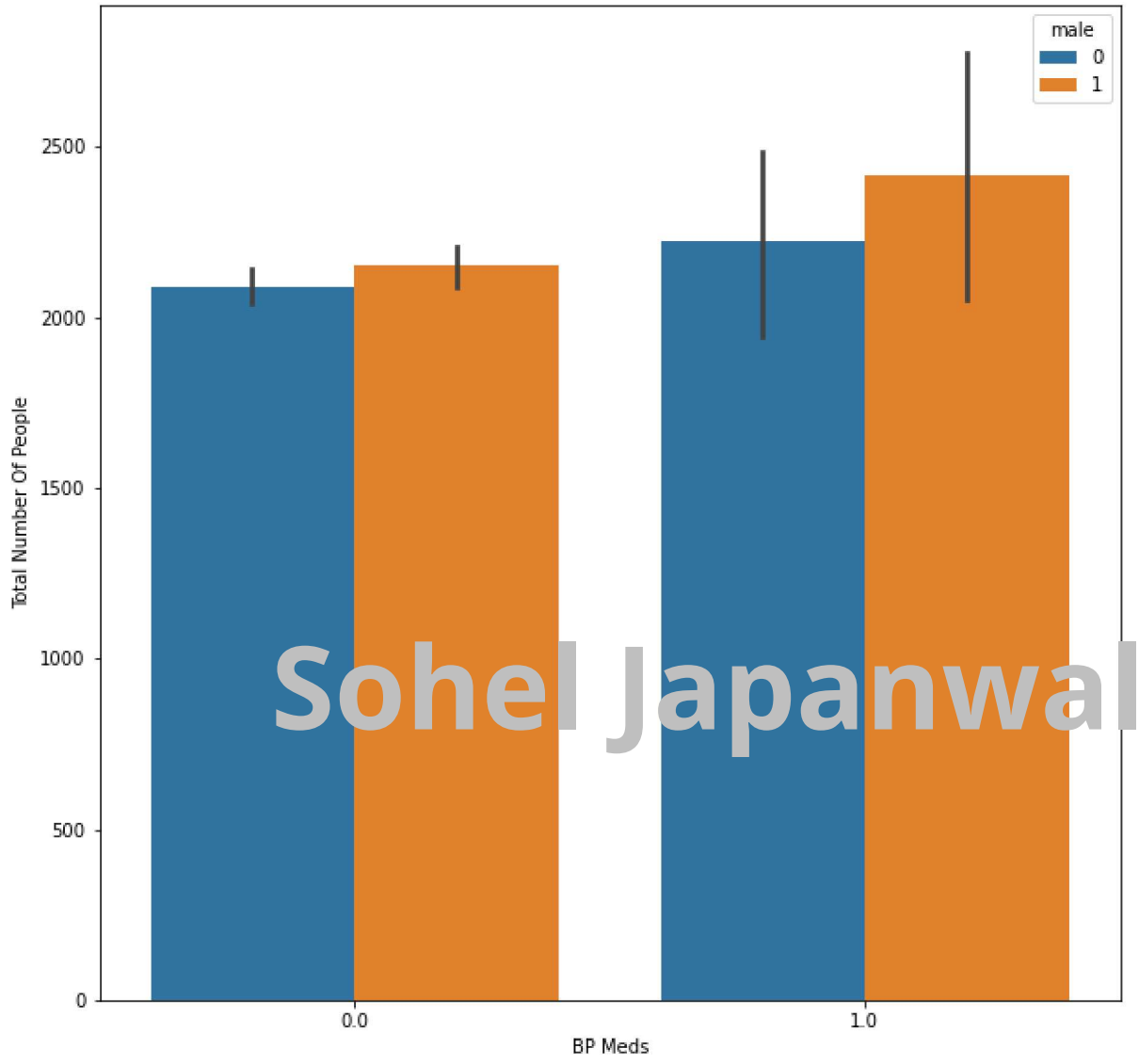
In [15]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["cigsPerDay"], hue=framingha
mDf["male"])
plt.ylabel("Total Cigarette Per Day")
plt.xlabel("Gender")
```

Out[15]: Text(0.5, 0, 'Gender')



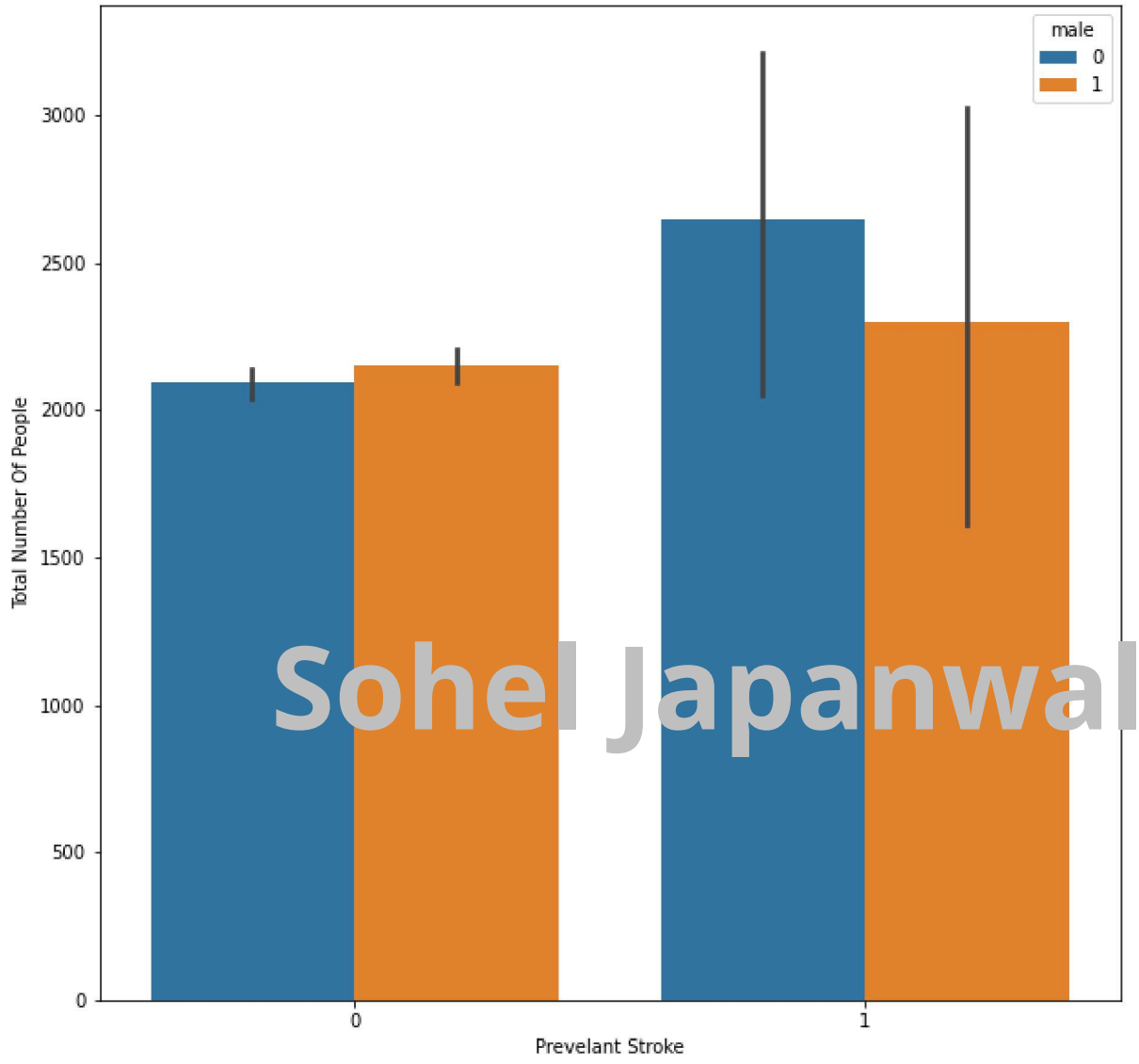
In [16]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["BPMeds"],y=framinghamDf["Serial"], hue=framinghamDf["male"])
plt.ylabel("Total Number Of People")
plt.xlabel("BP Meds")
```

Out[16]: Text(0.5, 0, 'BP Meds')



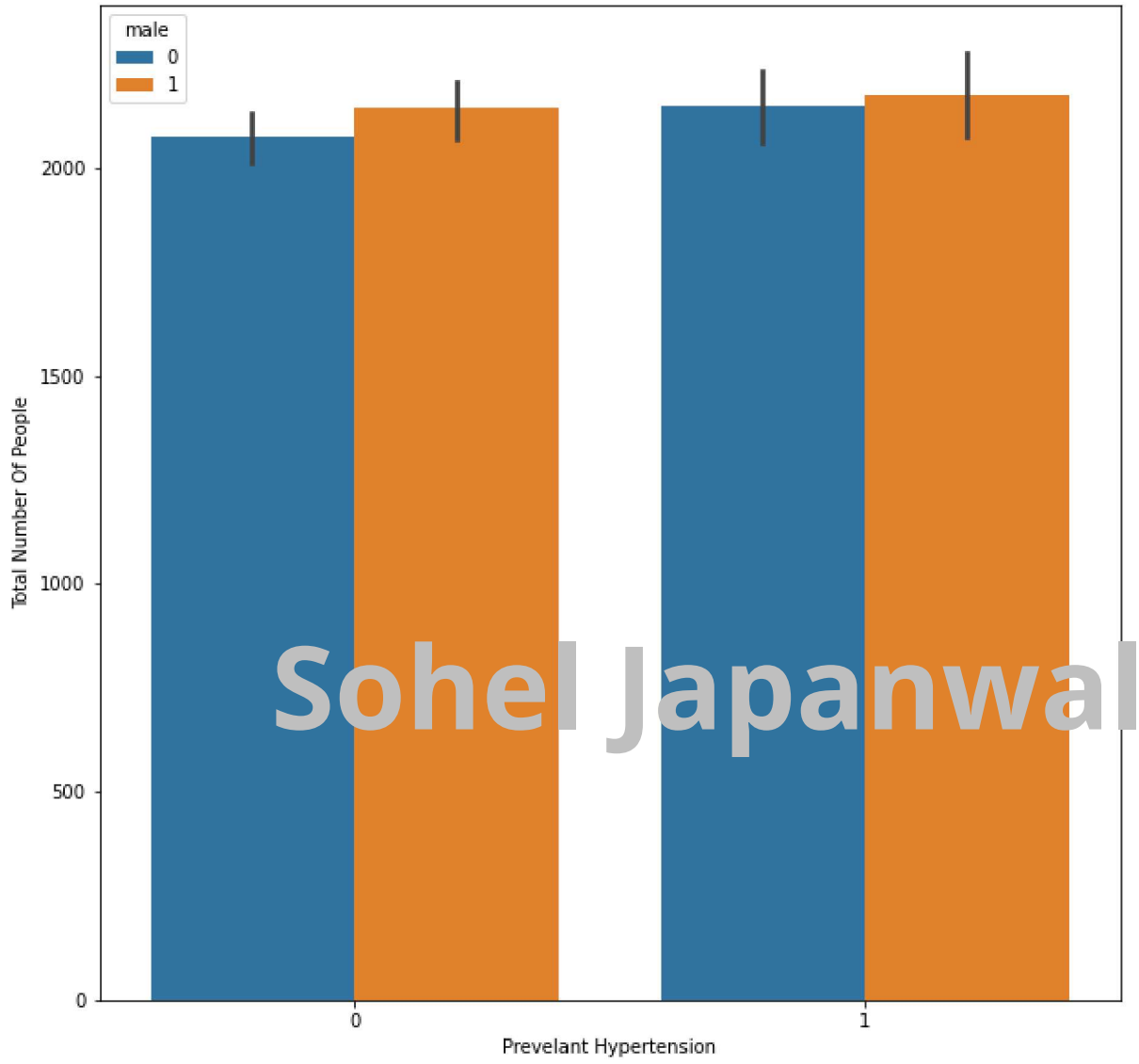
In [17]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["prevalentStroke"],y=framinghamDf["Serial"], hue=fr
aminghamDf["male"])
plt.ylabel("Total Number Of People")
plt.xlabel("Prevelant Stroke")
```

Out[17]: Text(0.5, 0, 'Prevelant Stroke')



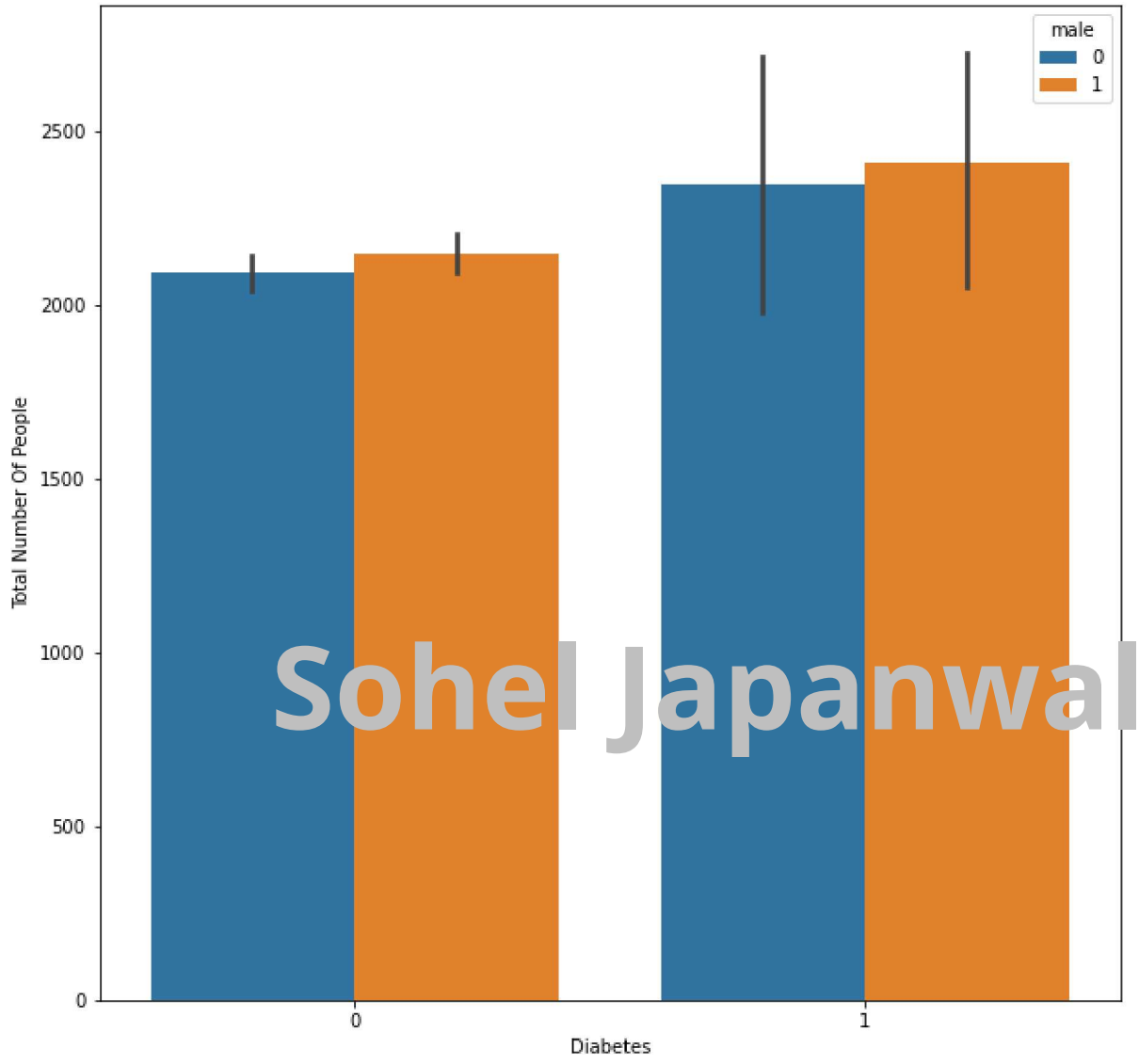
In [18]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["prevalentHyp"],y=framinghamDf["Serial"], hue=frami
nghamDf["male"])
plt.ylabel("Total Number Of People")
plt.xlabel("Prevelant Hypertension")
```

Out[18]: Text(0.5, 0, 'Prevelant Hypertension')



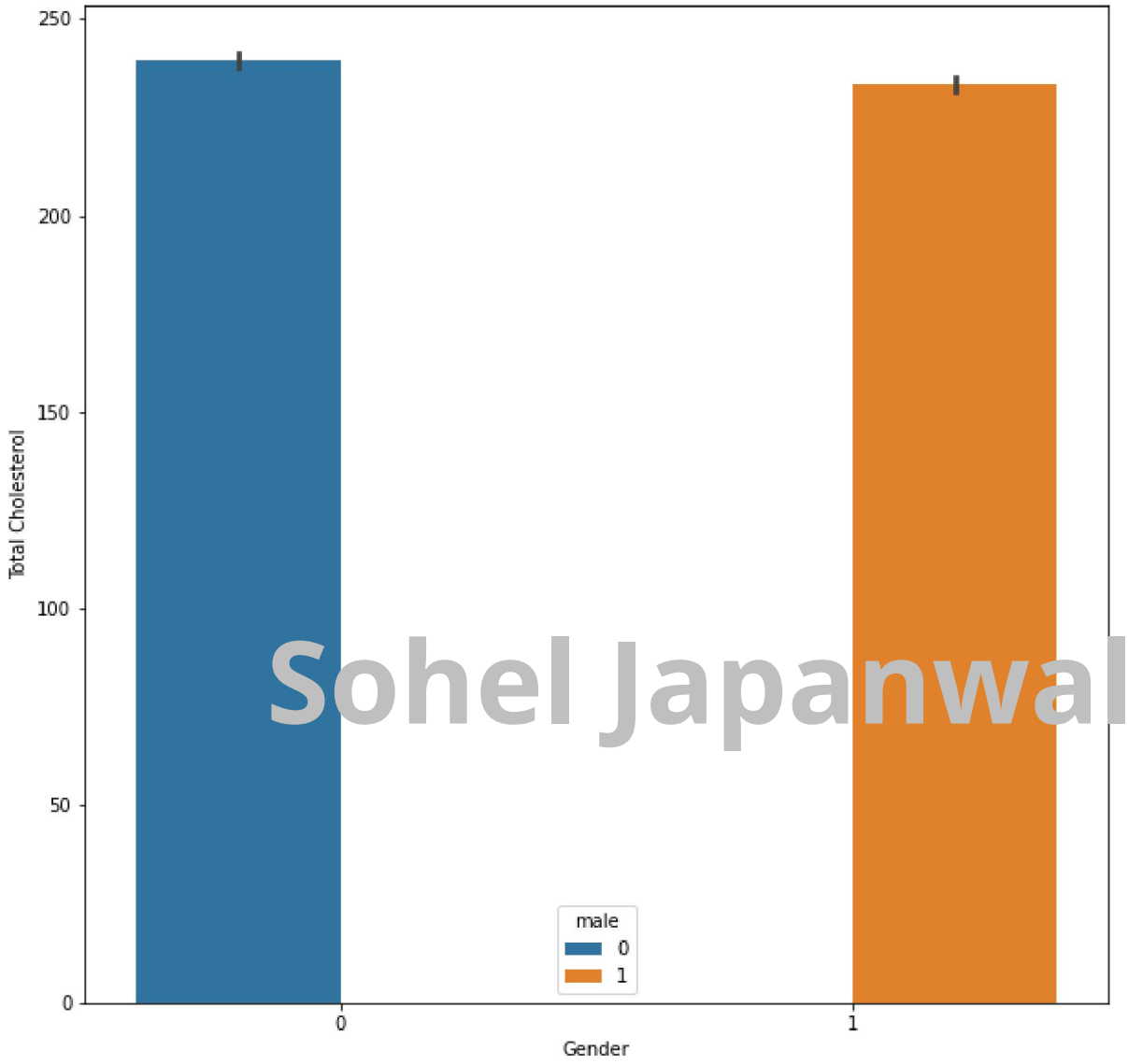
In [19]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["diabetes"],y=framinghamDf["Serial"], hue=framingha
mDf["male"])
plt.ylabel("Total Number Of People")
plt.xlabel("Diabetes")
```

Out[19]: Text(0.5, 0, 'Diabetes')



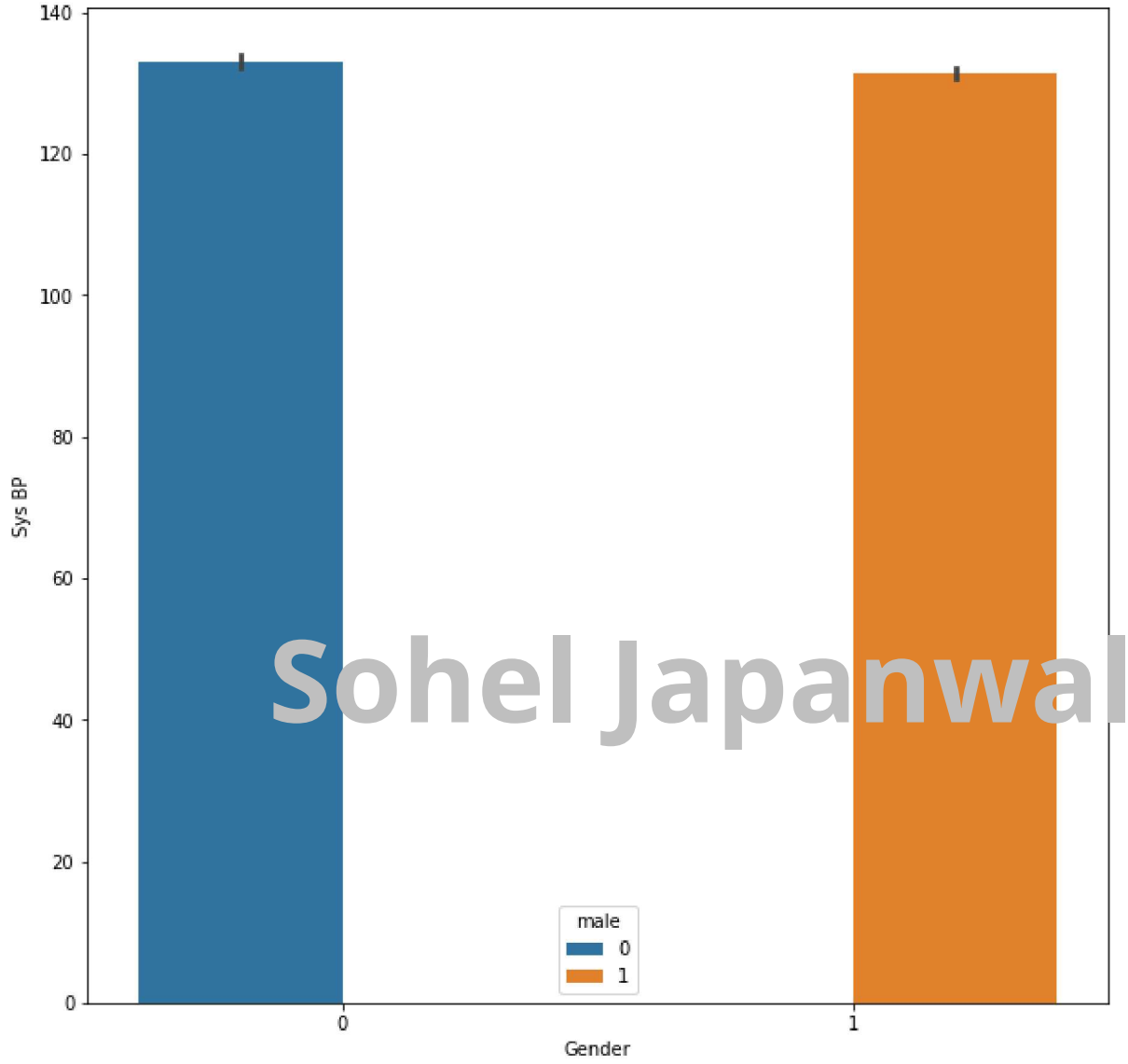
In [20]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["totChol"], hue=framinghamDf
["male"])
plt.ylabel("Total Cholesterol")
plt.xlabel("Gender")
```

Out[20]: Text(0.5, 0, 'Gender')



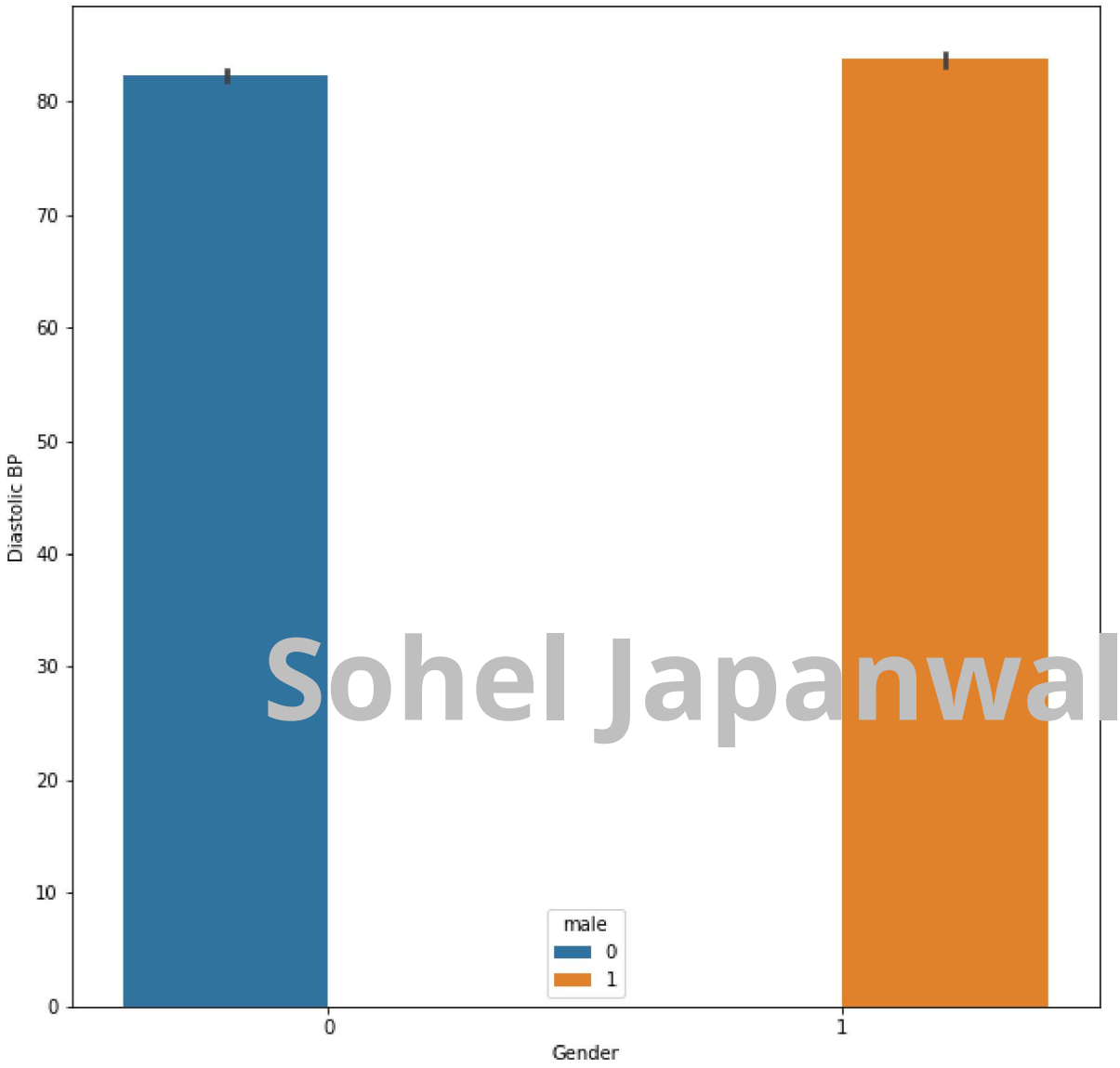
In [21]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["sysBP"], hue=framinghamDf[
"male"])
plt.ylabel("Sys BP")
plt.xlabel("Gender")
```

Out[21]: Text(0.5, 0, 'Gender')



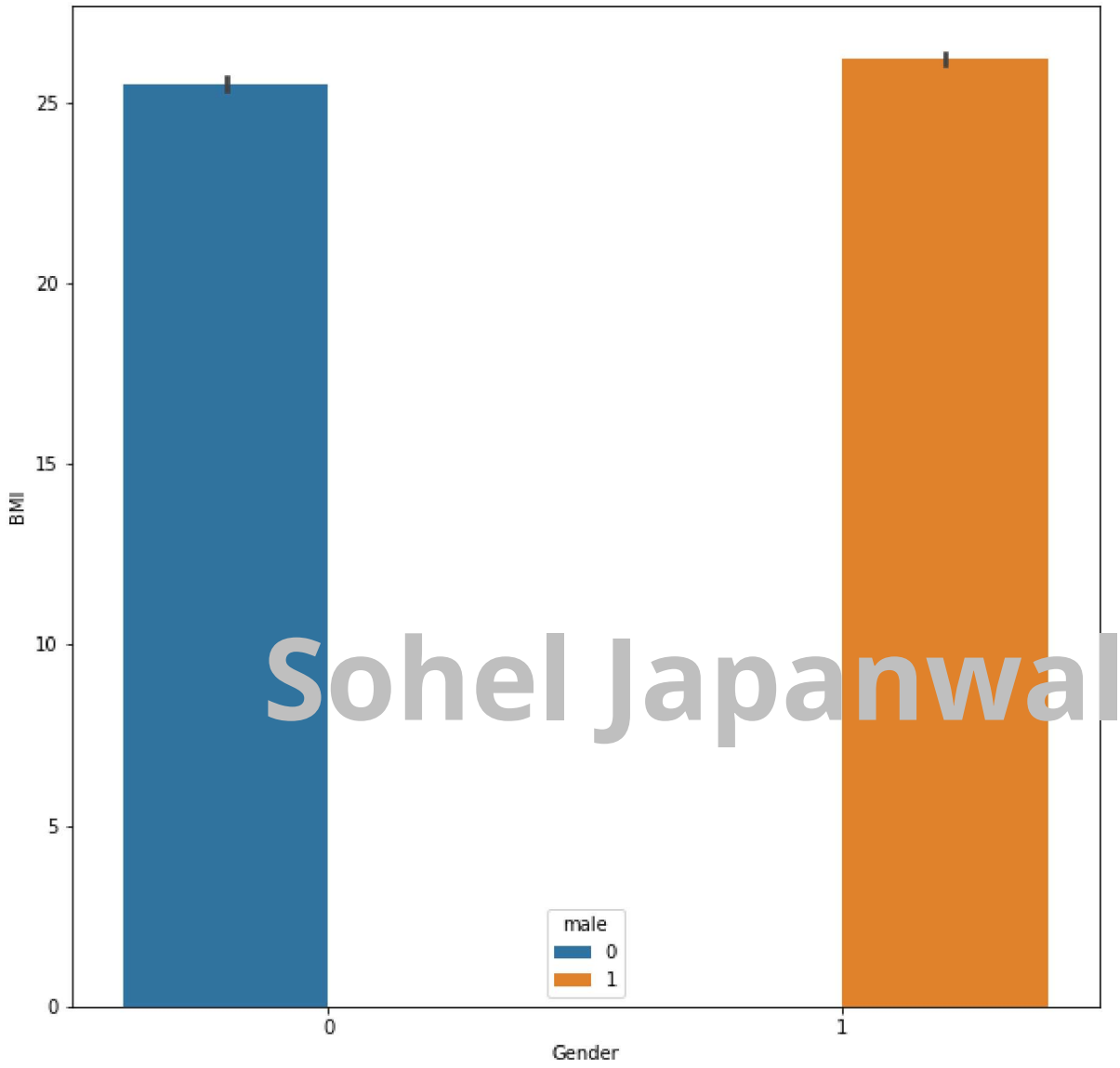
In [22]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["diaBP"], hue=framinghamDf[
"male"])
plt.ylabel("Diastolic BP")
plt.xlabel("Gender")
```

Out[22]: Text(0.5, 0, 'Gender')



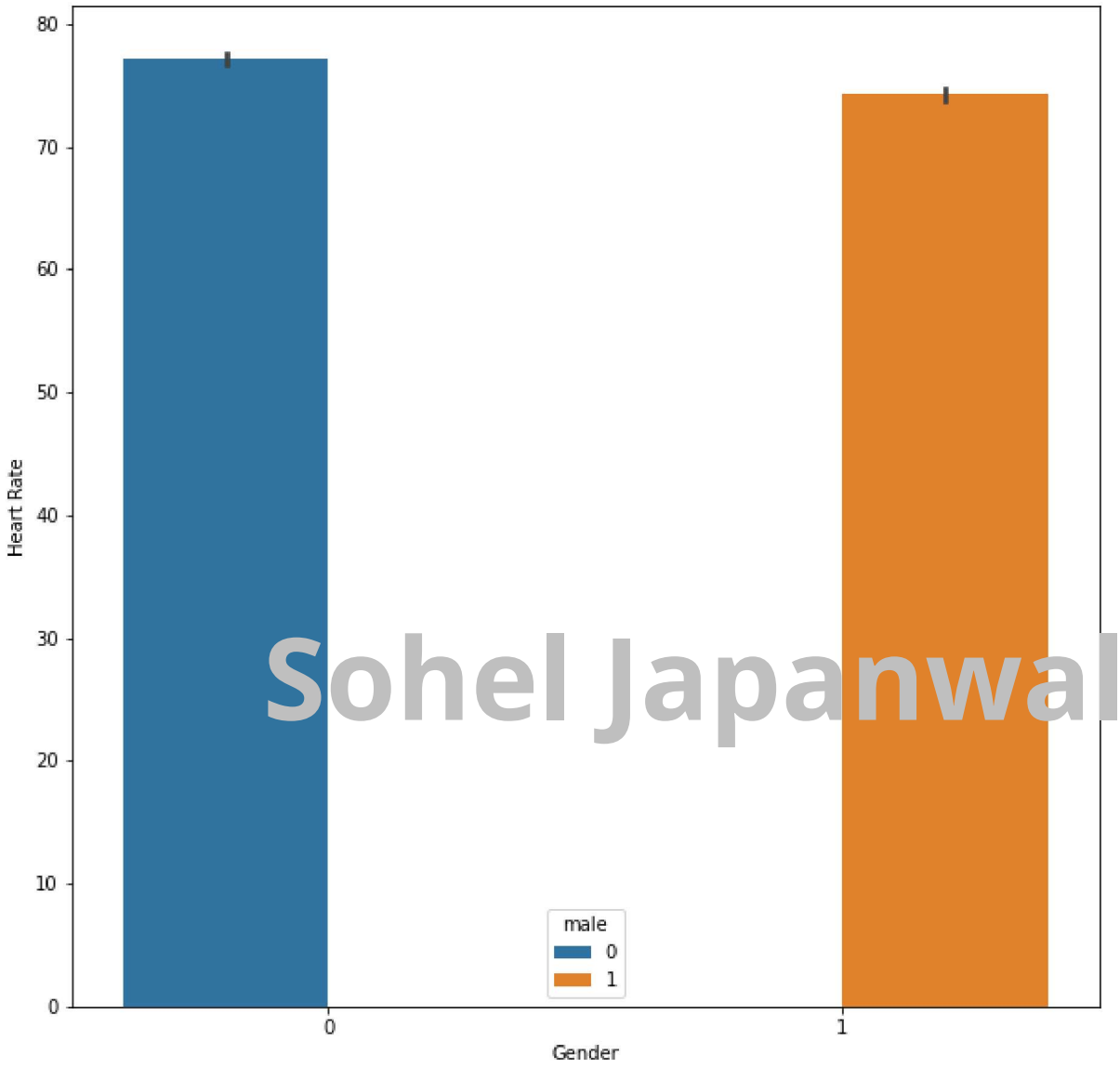
In [23]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["BMI"], hue=framinghamDf["male"])
plt.ylabel("BMI")
plt.xlabel("Gender")
```

Out[23]: Text(0.5, 0, 'Gender')



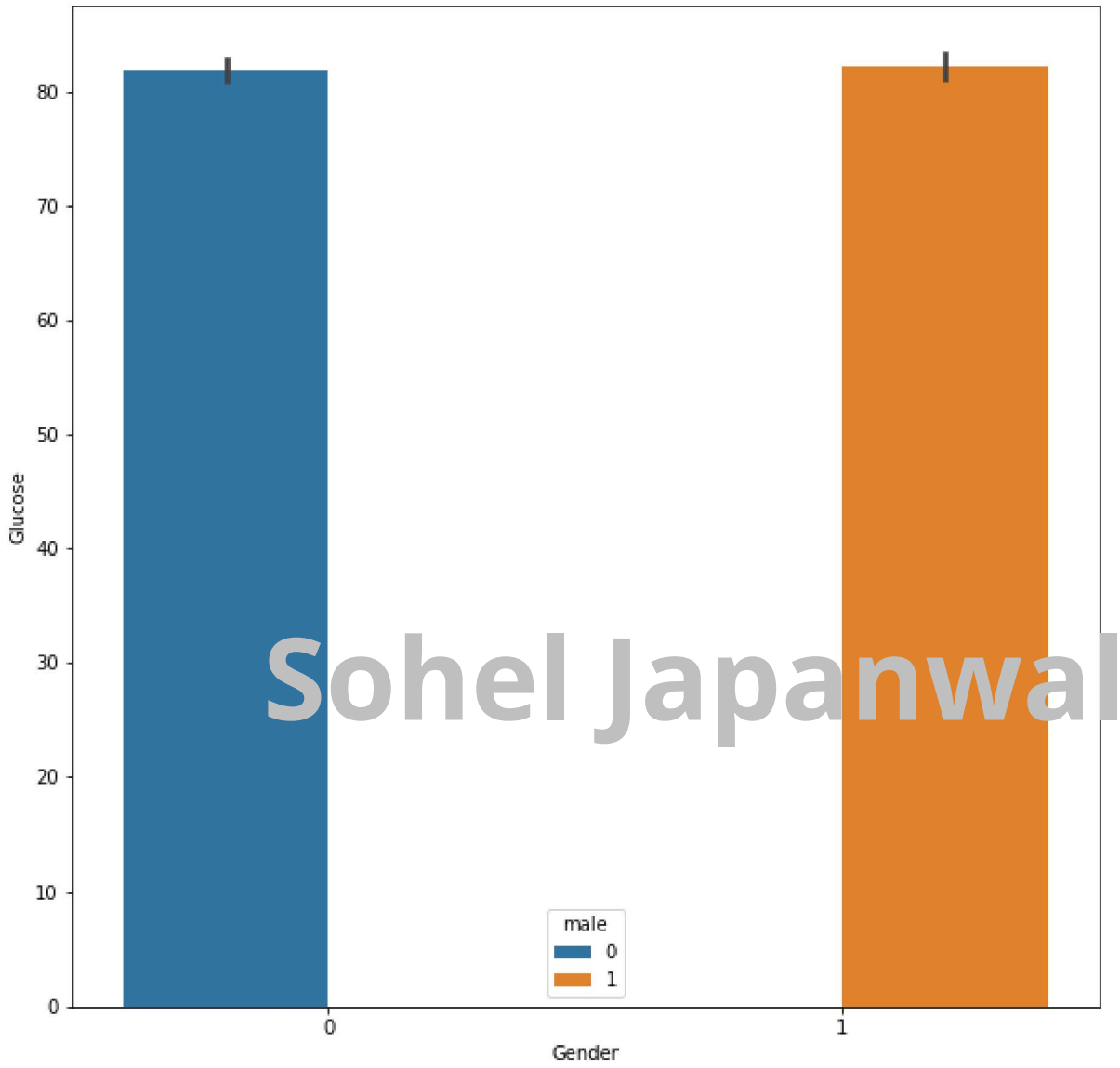
In [24]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["heartRate"], hue=framingham
Df["male"])
plt.ylabel("Heart Rate")
plt.xlabel("Gender")
```

Out[24]: Text(0.5, 0, 'Gender')



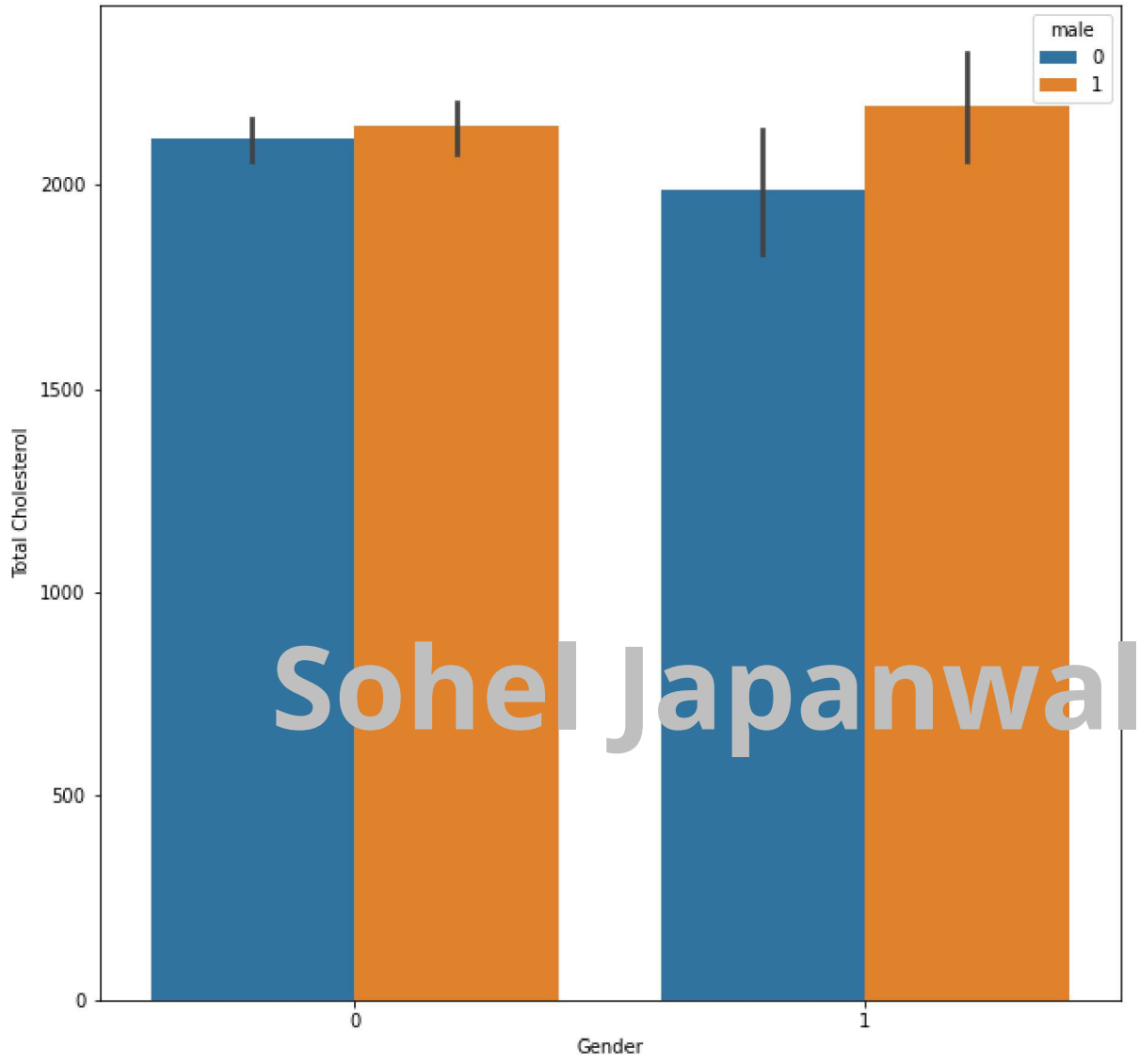
In [25]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["male"],y=framinghamDf["glucose"], hue=framinghamDf
["male"])
plt.ylabel("Glucose")
plt.xlabel("Gender")
```

Out[25]: Text(0.5, 0, 'Gender')



In [26]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=framinghamDf["TenYearCHD"],y=framinghamDf["Serial"], hue=framing
hamDf["male"])
plt.ylabel("Total Cholesterol")
plt.xlabel("Gender")
```

Out[26]: Text(0.5, 0, 'Gender')






In [27]:
```
framinghamDf["diabetes"].unique()
```

Out[27]: array([0, 1], dtype=int64)

In [28]:
```
framinghamDf.head()
```

Out[28]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | di |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 39 | 4.0 | 0 | 0.0 | 0.0 | 0 | 0 | |
| **1** | 0 | 46 | 2.0 | 0 | 0.0 | 0.0 | 0 | 0 | |
| **2** | 1 | 48 | 1.0 | 1 | 20.0 | 0.0 | 0 | 0 | |
| **3** | 0 | 61 | 3.0 | 1 | 30.0 | 0.0 | 0 | 1 | |
| **4** | 0 | 46 | 3.0 | 1 | 23.0 | 0.0 | 0 | 0 | |

## 4.2 Analyze Factors Affecting Health

In [30]:
```
framinghamDf.columns
```

Out[30]:
```
Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD', 'Serial'],
      dtype='object')
```

In [35]:
```
heartFactors=['male', 'age', 'currentSmoker', 'cigsPerDay',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD']
framinghamDf[heartFactors].corr()
```

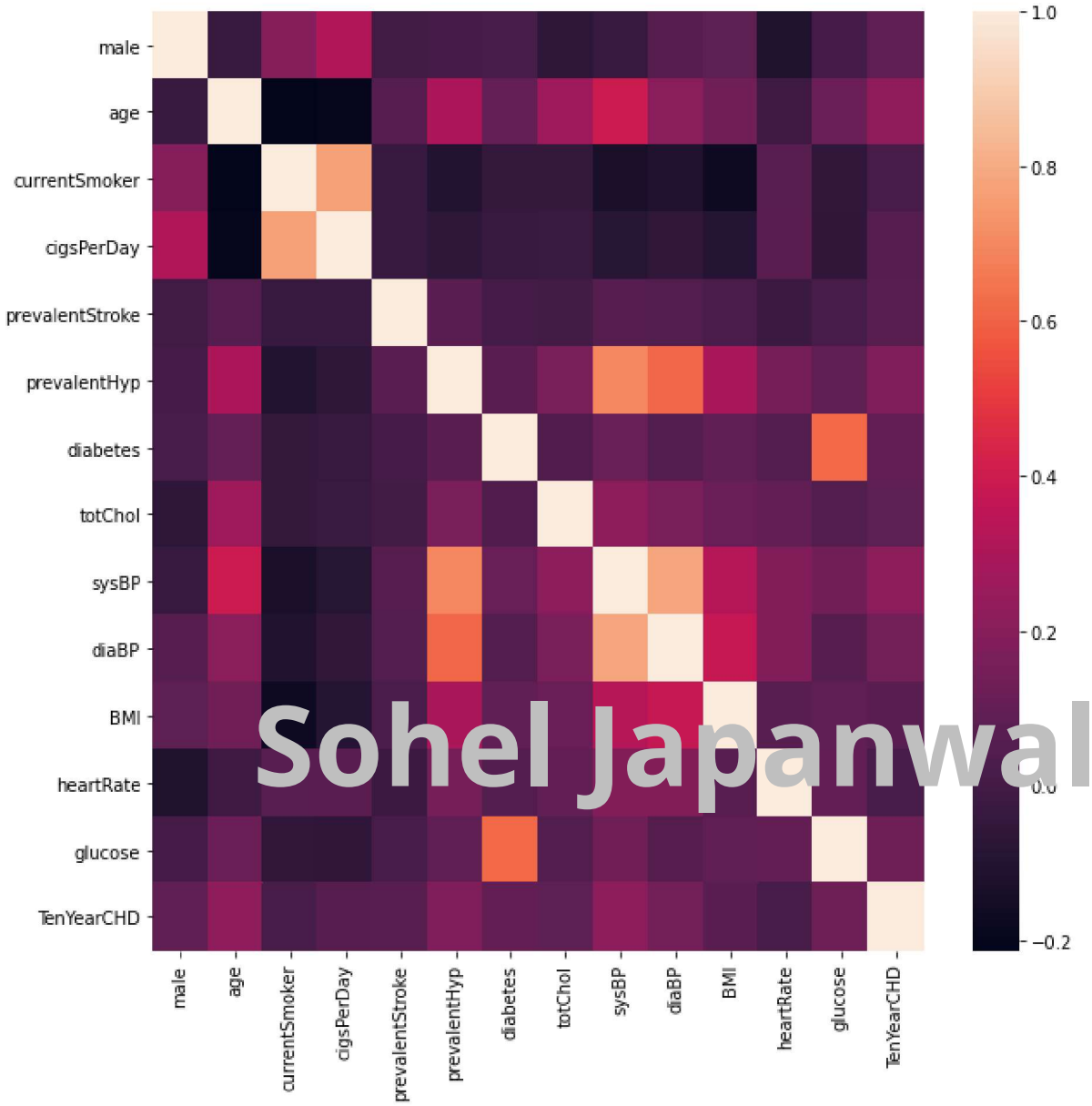


Out[35]:

| | male | age | currentSmoker | cigsPerDay | prevalentStroke | prevalentHyp |
|---|---|---|---|---|---|---|
| **male** | 1.000000 | -0.029014 | 0.197026 | 0.317143 | -0.004550 | 0.005853 |
| **age** | -0.029014 | 1.000000 | -0.213662 | -0.192959 | 0.057679 | 0.306799 |
| **currentSmoker** | 0.197026 | -0.213662 | 1.000000 | 0.769774 | -0.032980 | -0.103710 |
| **cigsPerDay** | 0.317143 | -0.192959 | 0.769774 | 1.000000 | -0.032711 | -0.066645 |
| **prevalentStroke** | -0.004550 | 0.057679 | -0.032980 | -0.032711 | 1.000000 | 0.074791 |
| **prevalentHyp** | 0.005853 | 0.306799 | -0.103710 | -0.066645 | 0.074791 | 1.000000 |
| **diabetes** | 0.015693 | 0.101314 | -0.044285 | -0.037089 | 0.006955 | 0.077752 |
| **totChol** | -0.070413 | 0.262554 | -0.046488 | -0.026479 | 0.000105 | 0.163632 |
| **sysBP** | -0.035879 | 0.394053 | -0.130281 | -0.088797 | 0.057000 | 0.696656 |
| **diaBP** | 0.058199 | 0.205586 | -0.107933 | -0.056715 | 0.045153 | 0.615840 |
| **BMI** | 0.081871 | 0.136096 | -0.167857 | -0.093293 | 0.025909 | 0.301344 |
| **heartRate** | -0.116932 | -0.012843 | 0.062686 | 0.075564 | -0.017674 | 0.146815 |
| **glucose** | 0.005979 | 0.122356 | -0.056726 | -0.058886 | 0.018440 | 0.086656 |
| **TenYearCHD** | 0.088374 | 0.225408 | 0.019448 | 0.057755 | 0.061823 | 0.177458 |

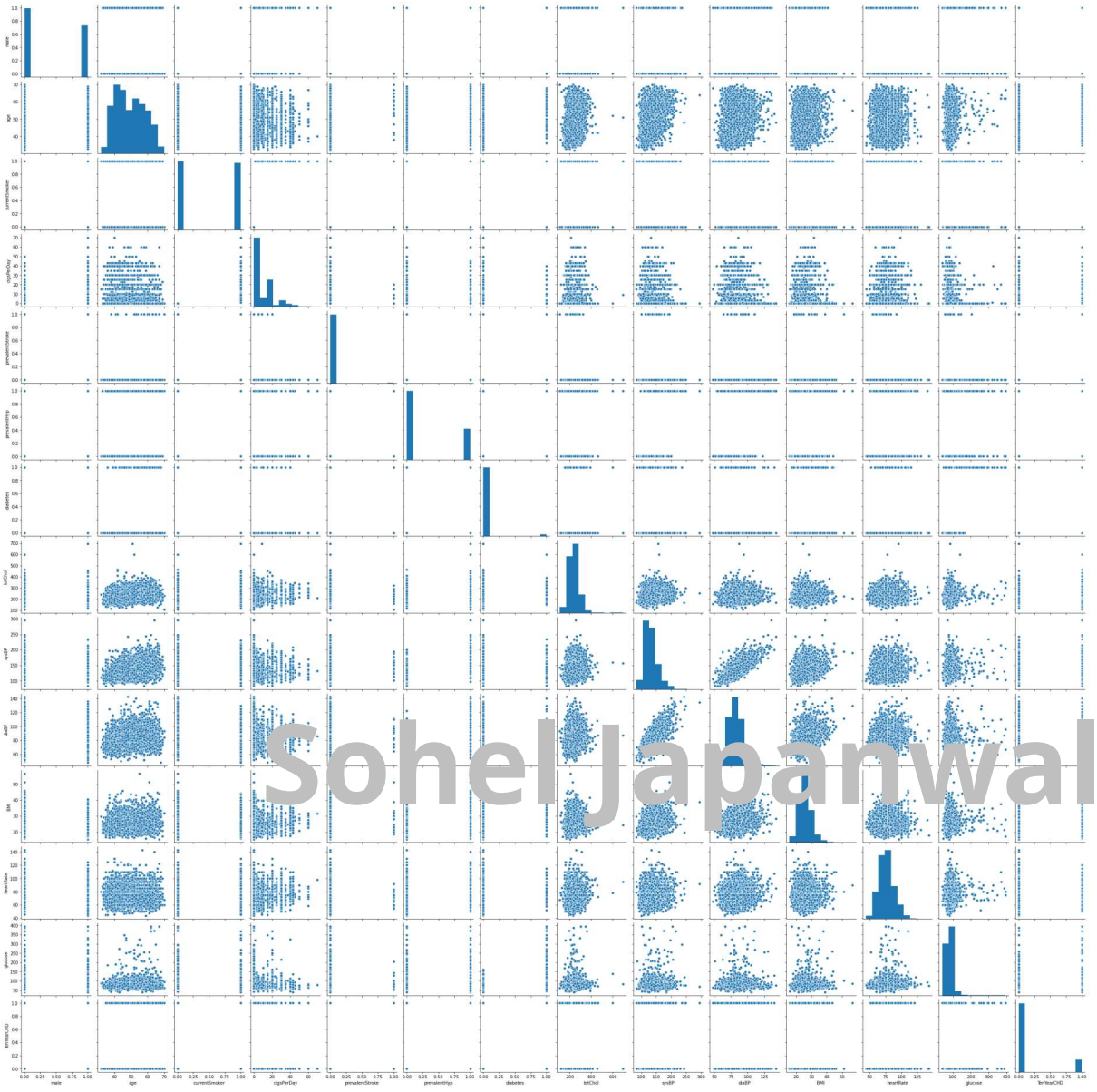
In [41]:
```
plt.figure(figsize=(10,10))
sns.heatmap(framinghamDf[heartFactors].corr())
```

Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x2b4b5e63790>



In [43]:
```
#plt.figure(figsize=(40,10))
sns.pairplot(framinghamDf[heartFactors])
```

Out[43]: <seaborn.axisgrid.PairGrid at 0x2b4e1c0cbe0>



## Observations:

With regards to **Heart health deterioration:**

- **Growing age has a very weak correlation(0.22)**
- **Hypertension has a very weak correlation(0.17)**
- **Systolic BP has a weak correlation(0.21)**
- **Diastolic BP has very weak correlation(0.14)**

With regards to **Glucose:**

- **Growing age has a very weak correlation(0.12)**
- **Diabetes has a moderate correlation(0.61)**
- **Systolic BP has a very weak correlation(0.14)**
- **Heart health has a very weak correlation(0.12)**

With regards to **Heart rate:**

- **Systolic BP has a very weak correlation(0.18)**
- **Diastolic BP has a very weak correlation(0.18)**

With regards to **BMI:**

- **Hypertension has a weak correlation(0.30)**
- **Systolic BP has a weak correlation(0.32)**
- **Diastolic BP has a weak correlation(0.3**
- **Cholesterol has a very weak correlation(0.1**



With regards to **Diastolic BP:**

- **Age has a very weak correlation(0.20)**
- **Hypertension has a moderate correlation(0.61)**
- **Systolic BP has a strong correlation(0.7)**
- **Cholesterol has a very weak correlation(0.16)**

With regards to **Hypertension:**

- **Age has a weak correlation(0.30)**
- **Systolic has a strong correlation(0.69)**
- **Diastolic BP has a strong correlation(0.61)**
- **Cholesterol has a very weak correlation(0.16)**

## 5. Actionable Insights

I suggest the following basis above observations:

- **To maintian heart health with growing age, hypertension needs to be kept in check**
- **Cholesterol needs to be maintained to prevent hypertension and heart health derterioration**
- **BMI needs to be regularly monitored to keep hypertension and cholesterol in check**

# Data Analysis Of Framingham Heart Study

**Author: Sohel Japanwala**

**Email: sohel.japanwala@gmail.com**

**LinkedIn: https://www.linkedin.com/in/soheljapanwala/ (https://www.linkedin.com/in/soheljapanwala/)**