In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
data = pd.read_csv("train.csv")
data.head()
```

Out[2]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	208500
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	181500
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	223500
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	140000
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	250000

5 rows × 81 columns

In [4]:
```
features=data[['GrLivArea','BedroomAbvGr','FullBath','HalfBath']]

features['TotalBath']=features['FullBath'] + 0.5*features['HalfBath']
features.head()
```

```
C:\Users\vibha\AppData\Local\Temp\ipykernel_11988\3780245161.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  features['TotalBath']=features['FullBath'] + 0.5*features['HalfBath']
```

Out[4]:

	GrLivArea	BedroomAbvGr	FullBath	HalfBath	TotalBath
0	1710	3	2	1	2.5
1	1262	3	2	0	2.0
2	1786	3	2	1	2.5
3	1717	3	1	0	1.0
4	2198	4	2	1	2.5

In [5]:
```
features = features.drop(columns=['FullBath','HalfBath'])
features.head()
```

Out[5]:

	GrLivArea	BedroomAbvGr	TotalBath
0	1710	3	2.5
1	1262	3	2.0
2	1786	3	2.5
3	1717	3	1.0
4	2198	4	2.5

In [6]:
```
target = data['SalePrice']
target.head()
```

Out[6]:
```
0    208500
1    181500
2    223500
3    140000
4    250000
Name: SalePrice, dtype: int64
```

In [7]:
```
features.isnull().sum()
```

Out[7]:
```
GrLivArea       0
BedroomAbvGr    0
TotalBath       0
dtype: int64
```

In [8]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
```

In [9]:
```
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)
```

In [10]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[10]:
```
▾ LinearRegression
LinearRegression()
```

In [11]:
```
y_pred = model.predict(X_test)
```

In [12]:
```
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
```

In [13]:
```
# User Input
x = float(input("House Sqt.: "))
y = int(input("No. of Bedrooms: "))
z = int(input("No. of washrooms: "))

# Combining input data into a tuple
input_data = (x, y, z)

# Changing the input_data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# Reshape the numpy array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)

# Making prediction
price = model.predict(input_data_reshaped)

print("House price for given Input data:", price)
```

```
House Sqt.: 1200
No. of Bedrooms: 3
No. of washrooms: 2
House price for given Input data: [151856.66969204]
C:\Users\vibha\Downloads\Anaconda\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

In [ ]: