In [21]:

```
import pandas as pd
```

In [22]:

```
df = pd.read_csv('house_data.csv')
```

In [23]:

```
print(df.head())
```

```
   id  NoOfRooms    Price
0   1          2  22000.0
1   2          3  33000.0
2   3          7  77000.0
3   4          2  22000.0
4   5          6  66000.0
```

In [24]:

```
print(df.isnull().sum())
```

```
id           0
NoOfRooms    0
Price        4
dtype: int64
```

In [25]:

```
print("\nHouses whose prices are more than $50,000:")
print(df[df['Price'] > 50000])
```

```

Houses whose prices are more than $50,000:
   id  NoOfRooms    Price
2   3          7  77000.0
4   5          6  66000.0
```

In [26]:

```
df = df.dropna(subset=['Price'])
print("\nUpdated dataset after removing null values in 'Price' column:")
print(df)
```

```

Updated dataset after removing null values in 'Price' column:
    id  NoOfRooms    Price
0    1          2  22000.0
1    2          3  33000.0
2    3          7  77000.0
3    4          2  22000.0
4    5          6  66000.0
5    6          3  33000.0
7    8          1  11000.0
9   10          3  33000.0
11  12          3  33000.0
13  14          2  22000.0
14  15          4  44000.0
```

In [27]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

In [28]:

```
X = df[['NoOfRooms']]
y = df['Price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, r
```

In [29]:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[29]:

```
▾ LinearRegression
LinearRegression()
```

In [30]:

```
y_pred = model.predict(X_test)
print("\nMean Squared Error: ", mean_squared_error(y_test, y_pred))
print("R-squared: ", r2_score(y_test, y_pred))
```

```

Mean Squared Error:  0.0
R-squared:  1.0
```

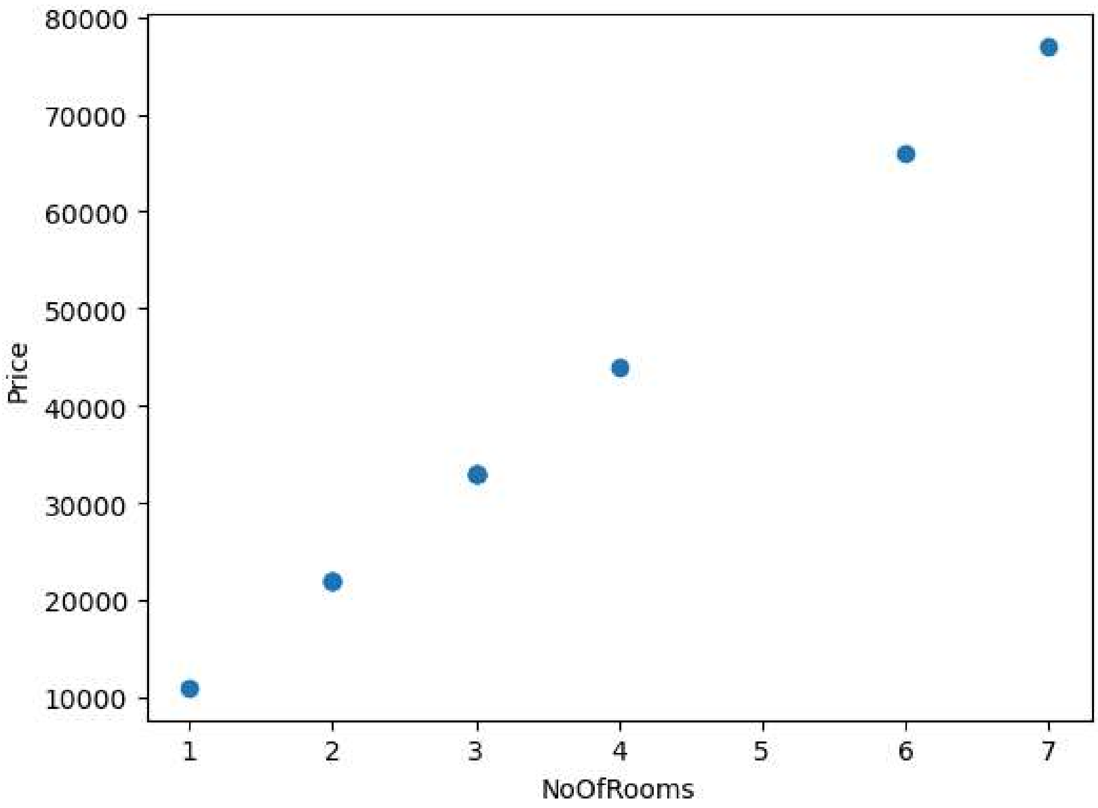
In [31]:

```
num_rooms = int(input("Enter the number of rooms: "))
new_house = pd.DataFrame({'NoOfRooms': [num_rooms]})
predicted_price = model.predict(new_house)
print("Predicted price for a house with", num_rooms, "rooms: ", predicted_
```

```
Enter the number of rooms: 5
Predicted price for a house with 5 rooms:  55000.0
```

In [32]:

```
plt.scatter(df['NoOfRooms'], df['Price'])
plt.xlabel('NoOfRooms')
plt.ylabel('Price')
plt.show()
```


Price
NoOfRooms