TASK 1

Implement a linear regression model to predict the prices of houses based on their square footage and the number of bedrooms and bathrooms.

In [3]:
```
import pandas as pd
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 [4]:
```
# Load the training dataset
train_data = pd.read_csv('/Users/manojt/Downloads/train.csv')

# Display the first few rows and check columns
print(train_data.head())
print(train_data.columns)
```

```
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotSha
pe  \
0   1          60       RL         65.0     8450   Pave   NaN      R
eg
1   2          20       RL         80.0     9600   Pave   NaN      R
eg
2   3          60       RL         68.0    11250   Pave   NaN      I
R1
3   4          70       RL         60.0     9550   Pave   NaN      I
R1
4   5          60       RL         84.0    14260   Pave   NaN      I
R1

  LandContour Utilities  ... PoolArea PoolQC Fence MiscFeature MiscV
al MoSold  \
0         Lvl    AllPub  ...        0    NaN   NaN         NaN
0      2
1         Lvl    AllPub  ...        0    NaN   NaN         NaN
0      5
2         Lvl    AllPub  ...        0    NaN   NaN         NaN
0      9
3         Lvl    AllPub  ...        0    NaN   NaN         NaN
0      2
4         Lvl    AllPub  ...        0    NaN   NaN         NaN
0     12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000

[5 rows x 81 columns]
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'St
reet',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'Bld
gType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'Yea
rRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVn
rType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQu
al',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'He
ating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrS
F',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath',
'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'G
arageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'G
arageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'Poo
lQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleT
```

```
ype',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [5]:
```
# Select relevant features for X (input features) and y (target variab
X_train = train_data[['GrLivArea', 'BedroomAbvGr', 'FullBath']]
y_train = train_data['SalePrice']
```

In [7]:
```
# Load the testing dataset
test_data = pd.read_csv('/Users/manojt/Downloads/test.csv')

# Display the first few rows and check columns
print(test_data.head())
print(test_data.columns)

# Select relevant features for X_test (use the same features as X_trai
X_test = test_data[['GrLivArea', 'BedroomAbvGr', 'FullBath']]
```

```
     Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotS
hape  \
0  1461          20       RH         80.0    11622   Pave   NaN
Reg
1  1462          20       RL         81.0    14267   Pave   NaN
IR1
2  1463          60       RL         74.0    13830   Pave   NaN
IR1
3  1464          60       RL         78.0     9978   Pave   NaN
IR1
4  1465         120       RL         43.0     5005   Pave   NaN
IR1

  LandContour Utilities  ... ScreenPorch PoolArea PoolQC  Fence Misc
Feature  \
0         Lvl    AllPub  ...         120        0    NaN  MnPrv
NaN
1         Lvl    AllPub  ...           0        0    NaN    NaN
Gar2
2         Lvl    AllPub  ...           0        0    NaN  MnPrv
NaN
3         Lvl    AllPub  ...           0        0    NaN    NaN
NaN
4         HLS    AllPub  ...         144        0    NaN    NaN
NaN

  MiscVal MoSold  YrSold  SaleType  SaleCondition
0       0      6    2010        WD         Normal
1   12500      6    2010        WD         Normal
2       0      3    2010        WD         Normal
3       0      6    2010        WD         Normal
4       0      1    2010        WD         Normal

[5 rows x 80 columns]
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'St
reet',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'Bld
gType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'Yea
rRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVn
rType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQu
al',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'He
ating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrS
F',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath',
'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'G
arageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'G
arageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'Poo
lQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleT
```

```
ype',
       'SaleCondition'],
      dtype='object')
```

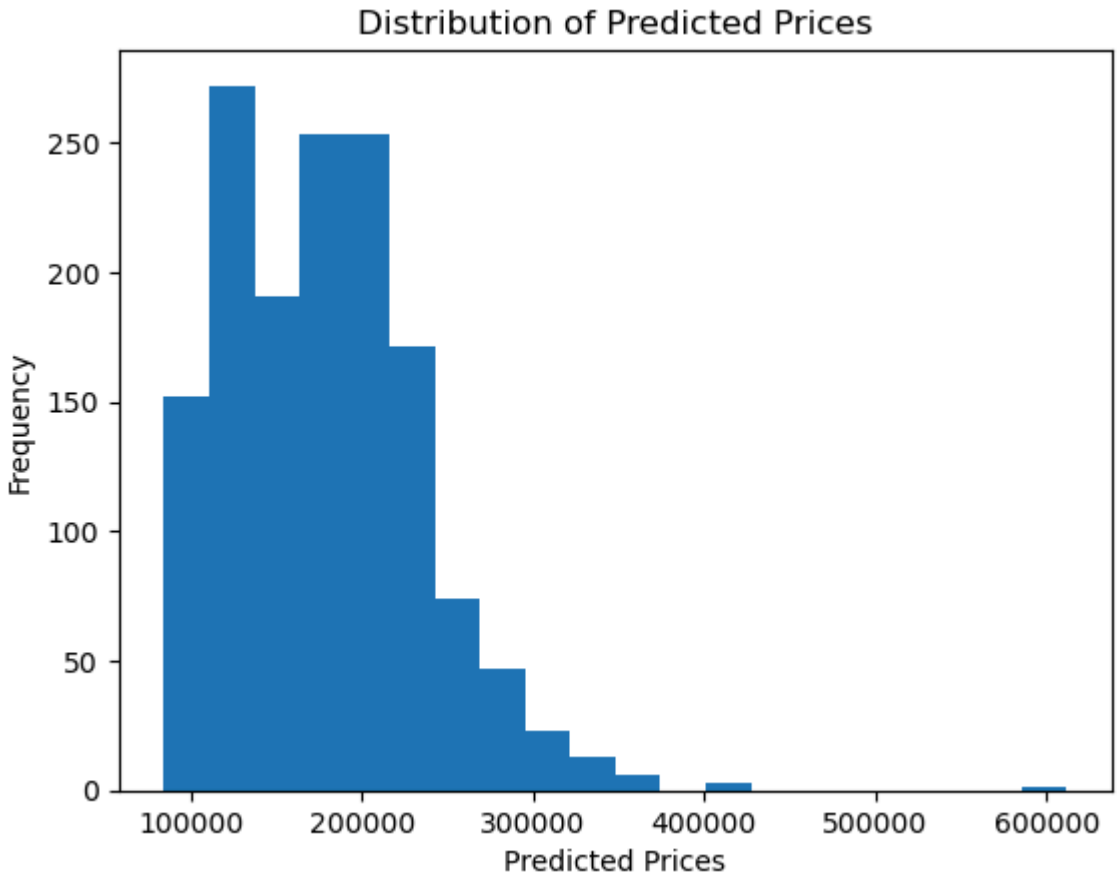
In [8]:
```
# Initialize the linear regression model
model = LinearRegression()

# Fit the model on the training data
model.fit(X_train, y_train)
```

Out[8]:
```
LinearRegression
LinearRegression()
```

In [9]:
```
# Predictions on the test set
y_pred = model.predict(X_test)
```

In [10]:
```
# Plotting predicted prices
plt.hist(y_pred, bins=20)
plt.xlabel('Predicted Prices')
plt.ylabel('Frequency')
plt.title('Distribution of Predicted Prices')
plt.show()
```



In [ ]: