```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

# Data as a dictionary
data = {
    'Area (sqft)': [1000, 1500, 1200, 1800, 1100, 1400, 2000, 1700, 1300, 2200, 1600, 1450, 950, 1850, 1250],
    'Bedrooms': [2, 3, 2, 4, 2, 3, 4, 3, 2, 5, 3, 3, 2, 4, 2],
    'Location': ['Chennai', 'Bangalore', 'Pune', 'Chennai', 'Bangalore', 'Pune', 'Chennai', 'Bangalore', 'Pune', 'Chennai', 'Bangalore', 'Pune
    'Price (Lakh)': [50, 75, 60, 90, 55, 70, 105, 85, 65, 120, 80, 72, 48, 95, 62]
}


def predict_house_price():
    """
    Trains a Linear Regression model to predict house prices and provides an
    interactive way to get predictions for new house details.
    """
    try:
        # 1. Load the housing dataset from the dictionary into a DataFrame
        df = pd.DataFrame(data)
        print("Dataset loaded successfully.")
        print("First 5 rows of the dataset:")
        print(df.head())
        print("-" * 30)

        # 2. Preprocessing and feature engineering
        # Separate features (X) and target (y)
        X = df.drop('Price (Lakh)', axis=1)
        y = df['Price (Lakh)']

        # Define which columns are numeric and which are categorical
        numeric_features = ['Area (sqft)', 'Bedrooms']
        categorical_features = ['Location']

        # Create a preprocessing pipeline
        # The ColumnTransformer applies different transformations to different columns
        preprocessor = ColumnTransformer(
            transformers=[
                ('cat', OneHotEncoder(), categorical_features)
            ],
            remainder='passthrough'
        )

        # 3. Split the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
        print(f"Data split into training ({len(X_train)} samples) and testing ({len(X_test)} samples).")
        print("-" * 30)

        # 4. Create and train the model using a pipeline
        # A pipeline chains together the preprocessor and the model, making the workflow cleaner
        model = Pipeline(steps=[
            ('preprocessor', preprocessor),
            ('regressor', LinearRegression())
        ])

        print("Training the Linear Regression model...")
        model.fit(X_train, y_train)
        print("Model training complete.")
        print("-" * 30)

        # 5. Evaluate the model on the test set
        y_pred = model.predict(X_test)

        r2 = r2_score(y_test, y_pred)
        mse = mean_squared_error(y_test, y_pred)

        print(f"Model Evaluation:")
        print(f"R-squared (R²) Score: {r2:.2f}")
        print(f"Mean Squared Error (MSE): {mse:.2f}")
        print("-" * 30)
        # 6. Predict house price based on user input
        print("Ready to predict house prices for new listings.")
        print("Example Input: Area: 1300, Bedrooms: 3, Location: Chennai")

        # Create a new DataFrame for user input
        new_data = pd.DataFrame([{
            'Area (sqft)': 1300,
            'Bedrooms': 3,
            'Location': 'Chennai'
        }])

        predicted_price = model.predict(new_data)

        print(f"\nPrediction for Area: 1300, Bedrooms: 3, Location: Chennai")
        print(f"Predicted House Price: {predicted_price[0]:.2f} Lakh")

    except FileNotFoundError:
        print(f"Error: 'house_data.csv' not found. Please create the file with the sample data.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Run the prediction function
if __name__ == "__main__":
    predict_house_price()
```

```
Dataset loaded successfully.
First 5 rows of the dataset:
   Area (sqft)  Bedrooms   Location  Price (Lakh)
0         1000         2    Chennai            50
1         1500         3  Bangalore            75
2         1200         2       Pune            60
3         1800         4    Chennai            90
4         1100         2  Bangalore            55
------------------------------
Data split into training (12 samples) and testing (3 samples).
------------------------------
Training the Linear Regression model...
Model training complete.
------------------------------
Model Evaluation:
R-squared (R²) Score: 0.98
Mean Squared Error (MSE): 15.03
------------------------------
Ready to predict house prices for new listings.
Example Input: Area: 1300, Bedrooms: 3, Location: Chennai

Prediction for Area: 1300, Bedrooms: 3, Location: Chennai
Predicted House Price: 65.96 Lakh
```