In [40]:
```
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LassoCV
from scipy import stats
import seaborn as sns
```

In [41]:
```
df = pd.read_csv("C:/Users/gundr/Downloads/Civil_Engineering_Regression_Dataset.csv")
```

In [44]:
```
X = df.drop(columns=["Project_ID", "Construction_Cost"])  # Exclude ID and target variable
Y = df["Construction_Cost"]
```

In [46]:
```
lasso = LassoCV(cv=5).fit(X, Y)
selected_features = X.columns[lasso.coef_ != 0]
X_selected = X[selected_features]
X_selected = sm.add_constant(X_selected)
```

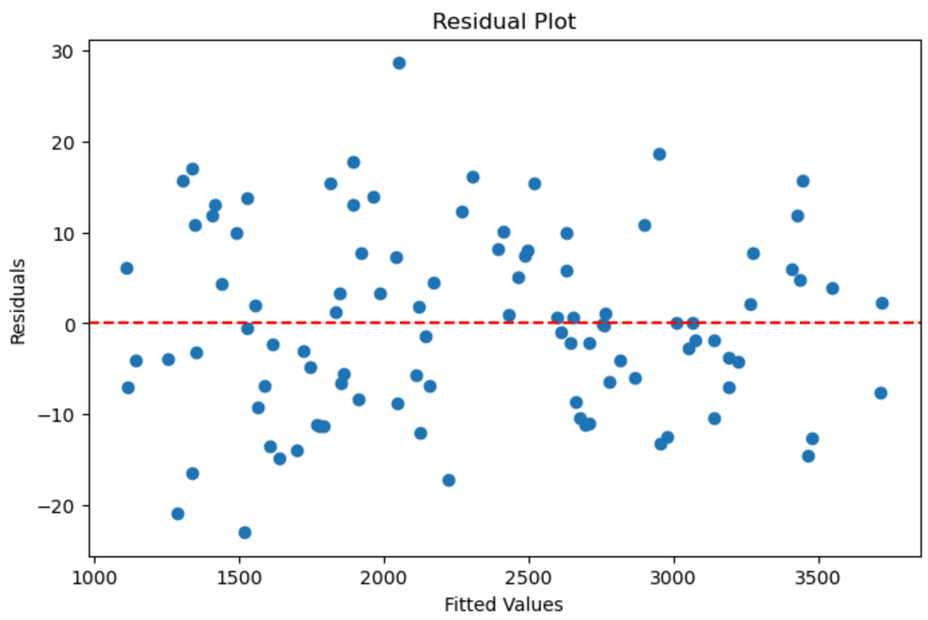
In [48]:
```
model = sm.OLS(Y, X_selected).fit()
```

In [50]:
```
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Construction_Cost   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 9.153e+04
Date:                Wed, 12 Feb 2025   Prob (F-statistic):          1.23e-171
Time:                        22:42:03   Log-Likelihood:                -372.31
No. Observations:                 100   AIC:                             756.6
Df Residuals:                      94   BIC:                             772.3
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                    -15.2800      6.217     -2.458      0.016     -27.624      -2.935
Building_Height           49.8898      0.079    628.588      0.000      49.732      50.047
Material_Quality_Index    10.6560      0.519     20.541      0.000       9.626      11.686
Labor_Cost                 0.5191      0.015     33.818      0.000       0.489       0.550
Concrete_Strength         20.3084      0.115    177.325      0.000      20.081      20.536
Foundation_Depth          30.0042      0.432     69.423      0.000      29.146      30.862
==============================================================================
Omnibus:                        1.217   Durbin-Watson:                   1.762
Prob(Omnibus):                  0.544   Jarque-Bera (JB):                1.258
Skew:                           0.186   Prob(JB):                        0.533
Kurtosis:                       2.596   Cond. No.                     1.23e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.23e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

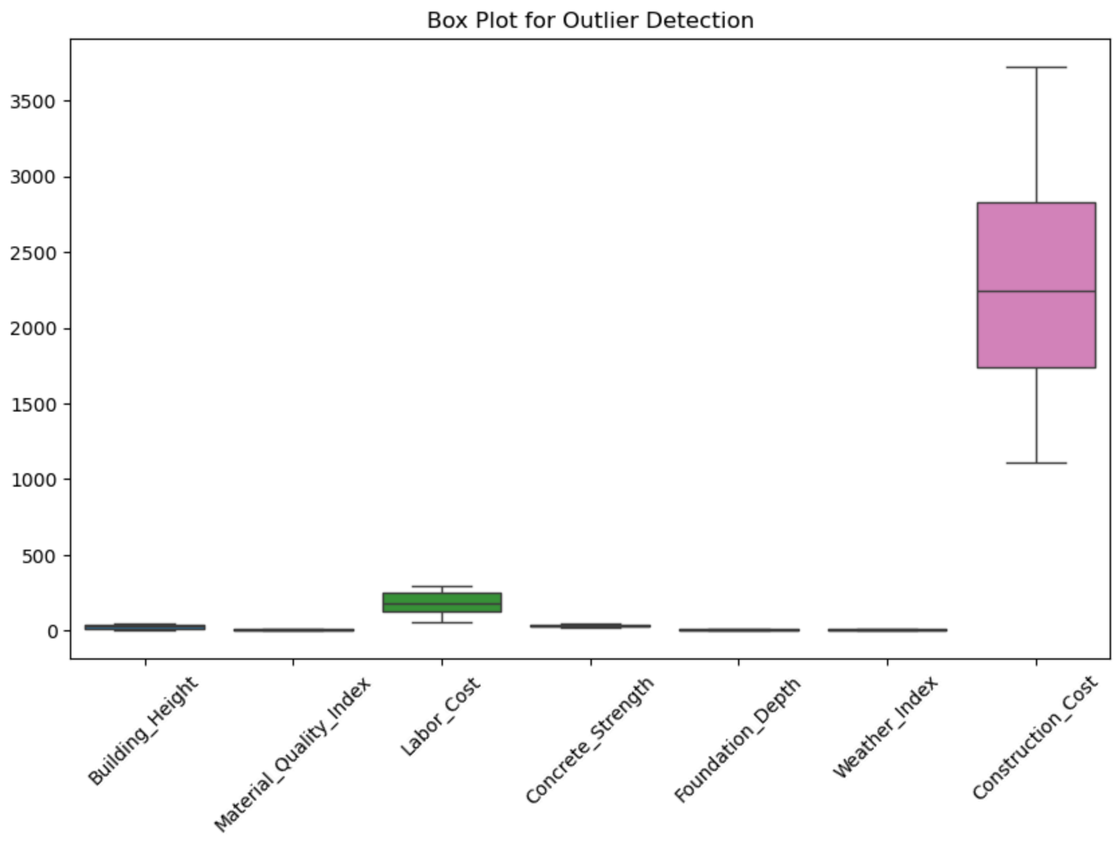
In [52]:
```
plt.figure(figsize=(8,5))
plt.scatter(model.fittedvalues, model.resid)
plt.axhline(y=0, color='r', linestyle='--')
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.title("Residual Plot")
plt.show()
```



In [54]:
```
z_scores = np.abs(stats.zscore(df.drop(columns=["Project_ID"])))
outliers = (z_scores > 3).any(axis=1)
print(f"Number of outliers detected: {outliers.sum()}")
```

```
Number of outliers detected: 0
```

In [56]:
```
plt.figure(figsize=(10,6))
sns.boxplot(data=df.drop(columns=["Project_ID"]))
plt.xticks(rotation=45)
plt.title("Box Plot for Outlier Detection")
plt.show()
```



In [58]:
```
print("\nModel Deployment Considerations:")
print("- Integrate real-time material and labor cost updates.")
print("- Incorporate location-based economic factors.")
print("- Utilize cloud-based APIs for dynamic cost estimation.")
```

```
Model Deployment Considerations:
- Integrate real-time material and labor cost updates.
- Incorporate location-based economic factors.
- Utilize cloud-based APIs for dynamic cost estimation.
```

In [60]:
```
print("\nEthical Considerations & Decision Making:")
print("- Overestimating costs may deter project investments.")
print("- Underestimating costs can lead to financial losses and project delays.")
print("- Accurate cost models enhance project safety by ensuring proper resource allocation.")
```

```
Ethical Considerations & Decision Making:
- Overestimating costs may deter project investments.
- Underestimating costs can lead to financial losses and project delays.
- Accurate cost models enhance project safety by ensuring proper resource allocation.
```

In [ ]:

In [ ]: