STEP 1: IMPORT THE NECESSARY LIBRARIES

In [21]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge, ElasticNet
from sklearn.metrics import mean_squared_error, r2_score
```

STEP 2: READ THE DATA FROM THE CSV FILES

- Dataset was downloaded from ourworldindata.org

In [22]:
```
df1 = pd.read_csv('mental-and-substance-use-as-share-of-disease.csv')
df2 = pd.read_csv('prevalence-by-mental-and-substance-use-disorder.csv')
```

STEP 3: FILL MISSING VALUES IN NUMERIC COLUMNS OF DATAFRAMES df1 AND df2 WITH THE MEAN OF THEIR RESPECTIVE COLUMNS

In [23]:
```
numeric_columns = df1.select_dtypes(include=[np.number]).columns
df1[numeric_columns] = df1[numeric_columns].fillna(df1[numeric_columns].m

numeric_columns = df2.select_dtypes(include=[np.number]).columns
df2[numeric_columns] = df2[numeric_columns].fillna(df2[numeric_columns].m
```

STEP 4: CONVERT DATA TYPES

In [24]:
```
df1['DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Bot
df2['Schizophrenia disorders (share of population) - Sex: Both - Age: Age
df2['Bipolar disorders (share of population) - Sex: Both - Age: Age-stand
df2['Eating disorders (share of population) - Sex: Both - Age: Age-standa
df2['Anxiety disorders (share of population) - Sex: Both - Age: Age-stand
df2['Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized
df2['Depressive disorders (share of population) - Sex: Both - Age: Age-st
df2['Prevalence - Alcohol use disorders - Sex: Both - Age: Age-standardiz
```

STEP 5: MERGE THE TWO DATAFRAMES ON A COMMON COLUMN

In [25]:
```
merged_df = pd.merge(df1, df2, on=['Entity', 'Code', 'Year'])
```

STEP 6: FEATURE THE MATRIX X AND THE VARIABLE y

In [26]:
```
X = merged_df[['Schizophrenia disorders (share of population) - Sex: Both
               'Bipolar disorders (share of population) - Sex: Both - Age
               'Eating disorders (share of population) - Sex: Both - Age:
               'Anxiety disorders (share of population) - Sex: Both - Age
               'Prevalence - Drug use disorders - Sex: Both - Age: Age-st
               'Depressive disorders (share of population) - Sex: Both -
               'Prevalence - Alcohol use disorders - Sex: Both - Age: Age

y = merged_df['DALYs (Disability-Adjusted Life Years) - Mental disorders
```

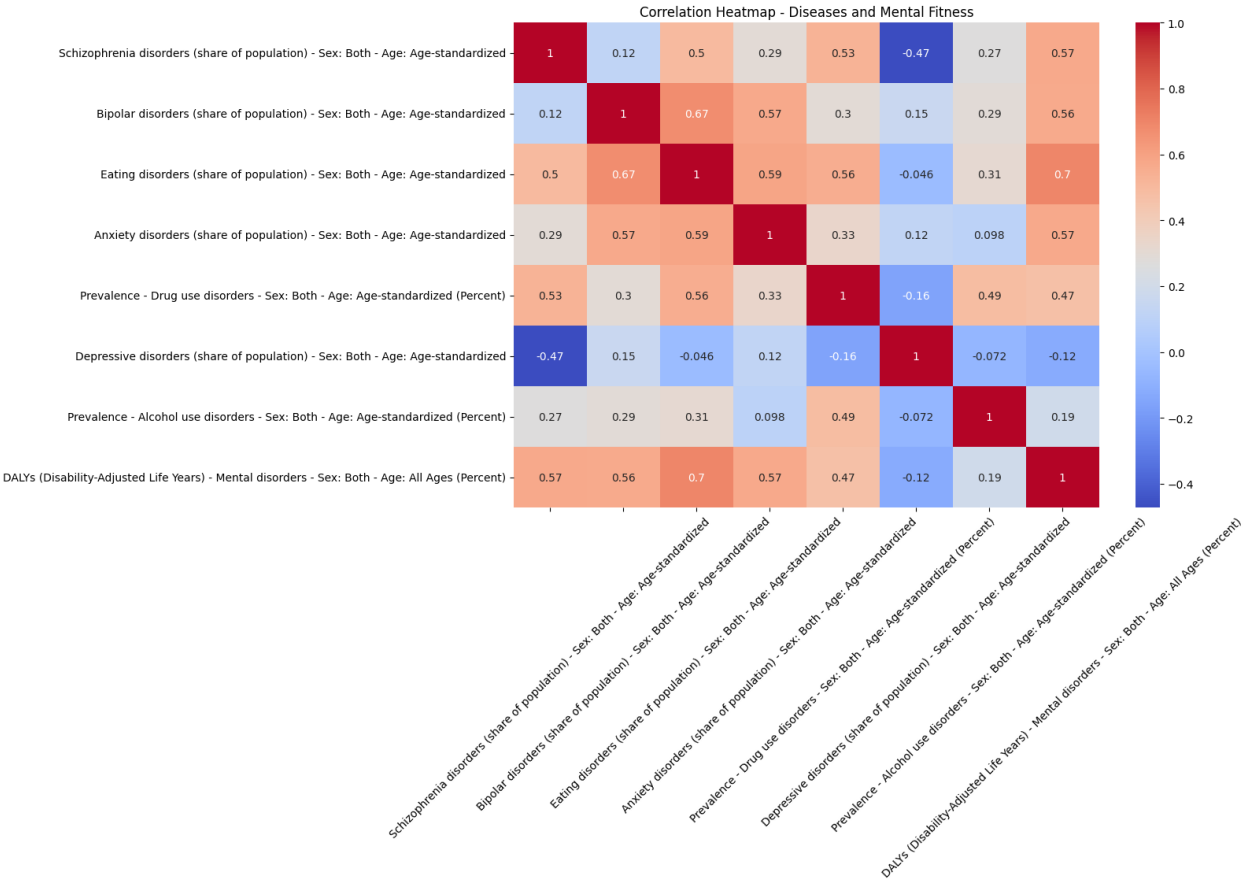
STEP 7: SPLIT THE DATA INTO TRAINING AND TESTING SETS

In [27]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
```

STEP 8: VISUALISING THE CORRELATION HEATMAP OF DISEASES AND MENTAL FITNESS

In [28]:
```
# Compute the correlation matrix
corr_matrix = merged_df[['Schizophrenia disorders (share of population) -
                         'Bipolar disorders (share of population) - Sex:
                         'Eating disorders (share of population) - Sex: B
                         'Anxiety disorders (share of population) - Sex:
                         'Prevalence - Drug use disorders - Sex: Both - A
                         'Depressive disorders (share of population) - Se
                         'Prevalence - Alcohol use disorders - Sex: Both
                         'DALYs (Disability-Adjusted Life Years) - Mental
                        ]].corr()

# Create the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap - Diseases and Mental Fitness')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.show()
```



STEP 9: FIT THE LINEAR REGRESSION MODEL

In [29]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[29]:
```
▾ LinearRegression
LinearRegression()
```

STEP 10: MAKE A PREDICTION USING TRAINED MODEL

In [30]:
```
y_pred = model.predict(X_test)
```

STEP 11: EVALUATE THE MODEL'S PERFORMANCE USING RIDGE REGRESSION AND LASSO REGRESSION

In [31]:
```
# Example using Ridge regression
from sklearn.linear_model import Ridge
ridge_model = Ridge(alpha=0.5)
ridge_model.fit(X_train, y_train)
ridge_y_pred = ridge_model.predict(X_test)
ridge_mse = mean_squared_error(y_test, ridge_y_pred)
ridge_r2 = r2_score(y_test, ridge_y_pred)
print('Ridge Regression - Mean Squared Error:', ridge_mse)
print('Ridge Regression - R-squared Score:', ridge_r2)

# Example using Elastic Net regression
from sklearn.linear_model import ElasticNet
elastic_net = ElasticNet(alpha=0.5, l1_ratio=0.5)  # Adjust the alpha and
elastic_net.fit(X_train, y_train)
elastic_net_y_pred = elastic_net.predict(X_test)
elastic_net_mse = mean_squared_error(y_test, elastic_net_y_pred)
elastic_net_r2 = r2_score(y_test, elastic_net_y_pred)
print('Elastic Net Regression - Mean Squared Error:', elastic_net_mse)
print('Elastic Net Regression - R-squared Score:', elastic_net_r2)
```

```
Ridge Regression - Mean Squared Error: 1.8852828652623428
Ridge Regression - R-squared Score: 0.6309285836156879
Elastic Net Regression - Mean Squared Error: 3.4451550539587945
Elastic Net Regression - R-squared Score: 0.325561018531185
```

STEP 12: FITTING REGRESSION MODELS TO THE TRAINING DATA AND MAKING PREDICTION ON THE TEST DATA AND CALCULATING MEAN SQUARED ERROR (MSE) AND R-SQUARED SCORE FOR EACH MODEL

In [32]:
```
# Fit Ridge Regression model
ridge_model = Ridge(alpha=1.0)
ridge_model.fit(X_train, y_train)

# Predict using Ridge Regression
ridge_y_pred = ridge_model.predict(X_test)

# Calculate MSE and R-squared score for Ridge Regression
ridge_mse = mean_squared_error(y_test, ridge_y_pred)
ridge_r2 = r2_score(y_test, ridge_y_pred)

print("Ridge Regression:")
print("Mean Squared Error (MSE):", ridge_mse)
print("R-squared Score:", ridge_r2)

# Fit Elastic Net Regression model
elastic_net_model = ElasticNet(alpha=1.0, l1_ratio=0.5)  # Adjust the alp
elastic_net_model.fit(X_train, y_train)

# Predict using Elastic Net Regression
elastic_net_y_pred = elastic_net_model.predict(X_test)

# Calculate MSE and R-squared score for Elastic Net Regression
elastic_net_mse = mean_squared_error(y_test, elastic_net_y_pred)
elastic_net_r2 = r2_score(y_test, elastic_net_y_pred)

print("Elastic Net Regression:")
print("Mean Squared Error (MSE):", elastic_net_mse)
print("R-squared Score:", elastic_net_r2)
```

```
Ridge Regression:
Mean Squared Error (MSE): 1.900177686158299
R-squared Score: 0.6280127067750545
Elastic Net Regression:
Mean Squared Error (MSE): 3.9995458450448202
R-squared Score: 0.21703099459333586
```

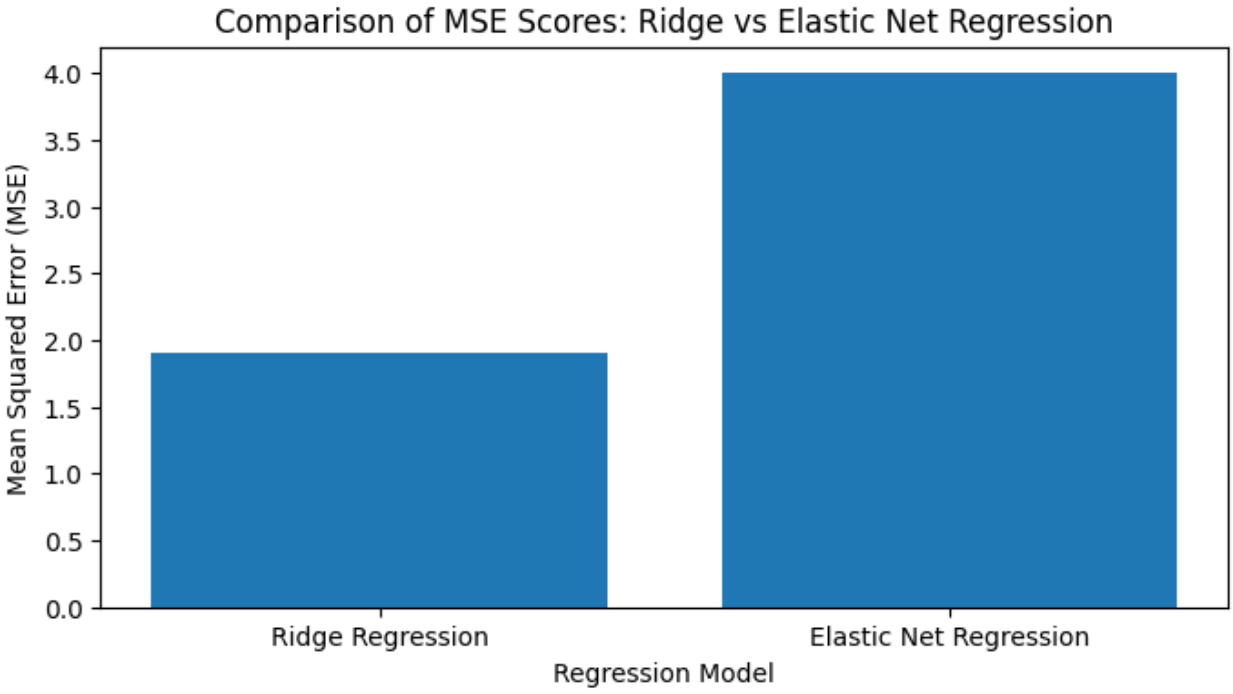
STEP 13: IT DISPLAYS SCORE OF BOTH THE MODELS IN A VISUAL FORMAT

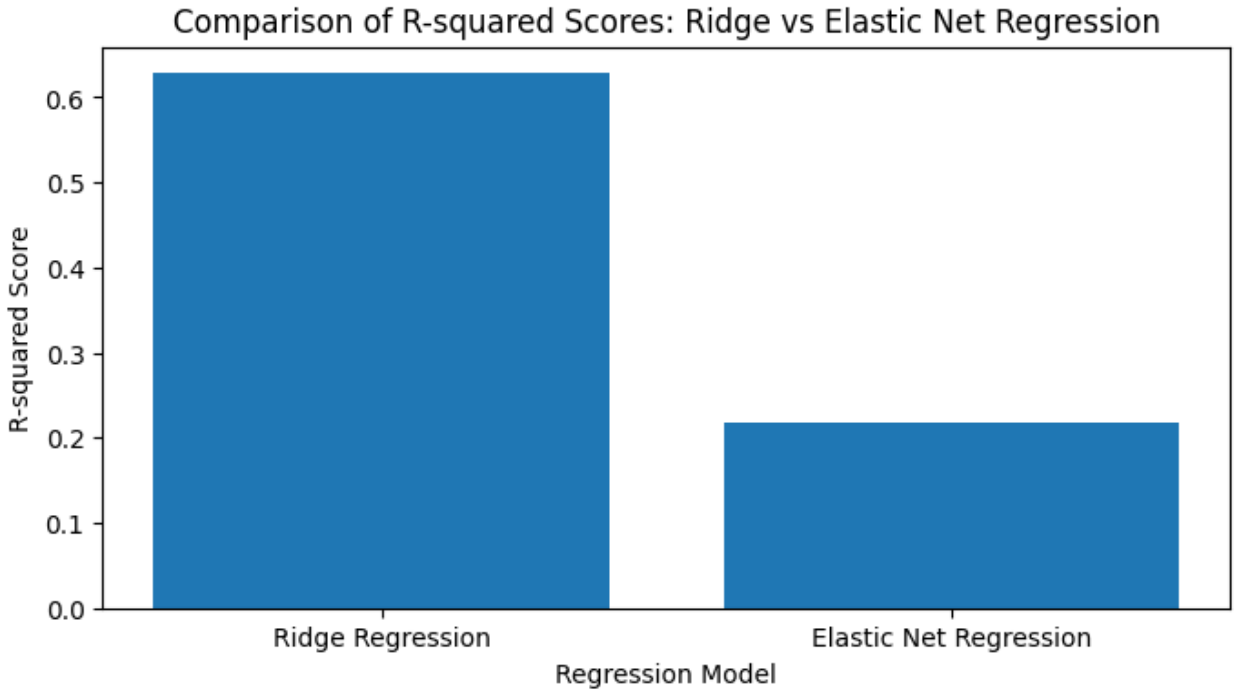
In [33]:
```
# Calculate MSE and R-squared scores
ridge_mse = mean_squared_error(y_test, ridge_y_pred)
ridge_r2 = r2_score(y_test, ridge_y_pred)

elastic_net_mse = mean_squared_error(y_test, elastic_net_y_pred)
elastic_net_r2 = r2_score(y_test, elastic_net_y_pred)

# Create a bar plot for MSE scores
plt.figure(figsize=(8, 4))
plt.bar(['Ridge Regression', 'Elastic Net Regression'], [ridge_mse, elast
plt.xlabel('Regression Model')
plt.ylabel('Mean Squared Error (MSE)')
plt.title('Comparison of MSE Scores: Ridge vs Elastic Net Regression')
plt.show()

# Create a bar plot for R-squared scores
plt.figure(figsize=(8, 4))
plt.bar(['Ridge Regression', 'Elastic Net Regression'], [ridge_r2, elasti
plt.xlabel('Regression Model')
plt.ylabel('R-squared Score')
plt.title('Comparison of R-squared Scores: Ridge vs Elastic Net Regressio
plt.show()
```





STEP 14: IT PRINTS OUT THE RESULT AS WELL AS THE CODE

In [35]:
```
# Compare the scores of Ridge Regression and Elastic Net Regression
if ridge_mse < elastic_net_mse and ridge_r2 > elastic_net_r2:
    print("Ridge Regression is more accurate.")
    print("Reason: Ridge Regression has a lower MSE and higher R-squared
elif elastic_net_mse < ridge_mse and elastic_net_r2 > ridge_r2:
    print("Elastic Net Regression is more accurate.")
    print("Reason: Elastic Net Regression has a lower MSE and higher R-sq
elif ridge_mse == elastic_net_mse and ridge_r2 == elastic_net_r2:
    print("Both Ridge Regression and Elastic Net Regression have similar
    print("Reason: The MSE and R-squared scores are equal for both models
else:
    print("Unable to determine which model is more accurate.")
```

```
Ridge Regression is more accurate.
Reason: Ridge Regression has a lower MSE and higher R-squared score.
```