In [1]:
```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

file_path = r"C:\Users\Yi Jun Zhuo\Downloads\diabetes.csv"
df = pd.read_csv(file_path)
```

In [2]:
```
#replace the data is 0 to median
cols = ['Glucose', 'BloodPressure', 'SkinThickness','Insulin', 'BMI']
df[cols] = df[cols].replace(0, np.nan)
df.fillna(df.median(),inplace=True)
print("\nMissing values after cleaninng:\n", df.isnull(). sum())
```

```
Missing values after cleaninng:
 Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [3]:
```
#create new column
df['BMI_Age'] = df['BMI']*df['Age']
df['Glucose_Insulin_ratio'] = df['Glucose']/(df['Insulin']+1)

print("\nNew features:\n",df[['BMI_Age', 'Glucose_Insulin_ratio']].head())
```

```
New features:
    BMI_Age  Glucose_Insulin_ratio
0   1680.0               1.174603
1    824.6               0.674603
2    745.6               1.452381
3    590.1               0.936842
4   1422.3               0.810651
```

In [4]:
```
#Select test and train group
X = df.drop('Outcome', axis =1)
Y = df['Outcome']
X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.2, random_state=42, stratify=Y
)

print("\nTraining size:", X_train.shape)
print("Test size:", X_test.shape)
```

```
Training size: (614, 10)
Test size: (154, 10)
```

In [5]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)\

model = DecisionTreeClassifier(
    max_depth=4, min_samples_split=20, class_weight='balanced',random_state=42
)
model.fit(X_train_scaled, Y_train)
```

Out[5]:

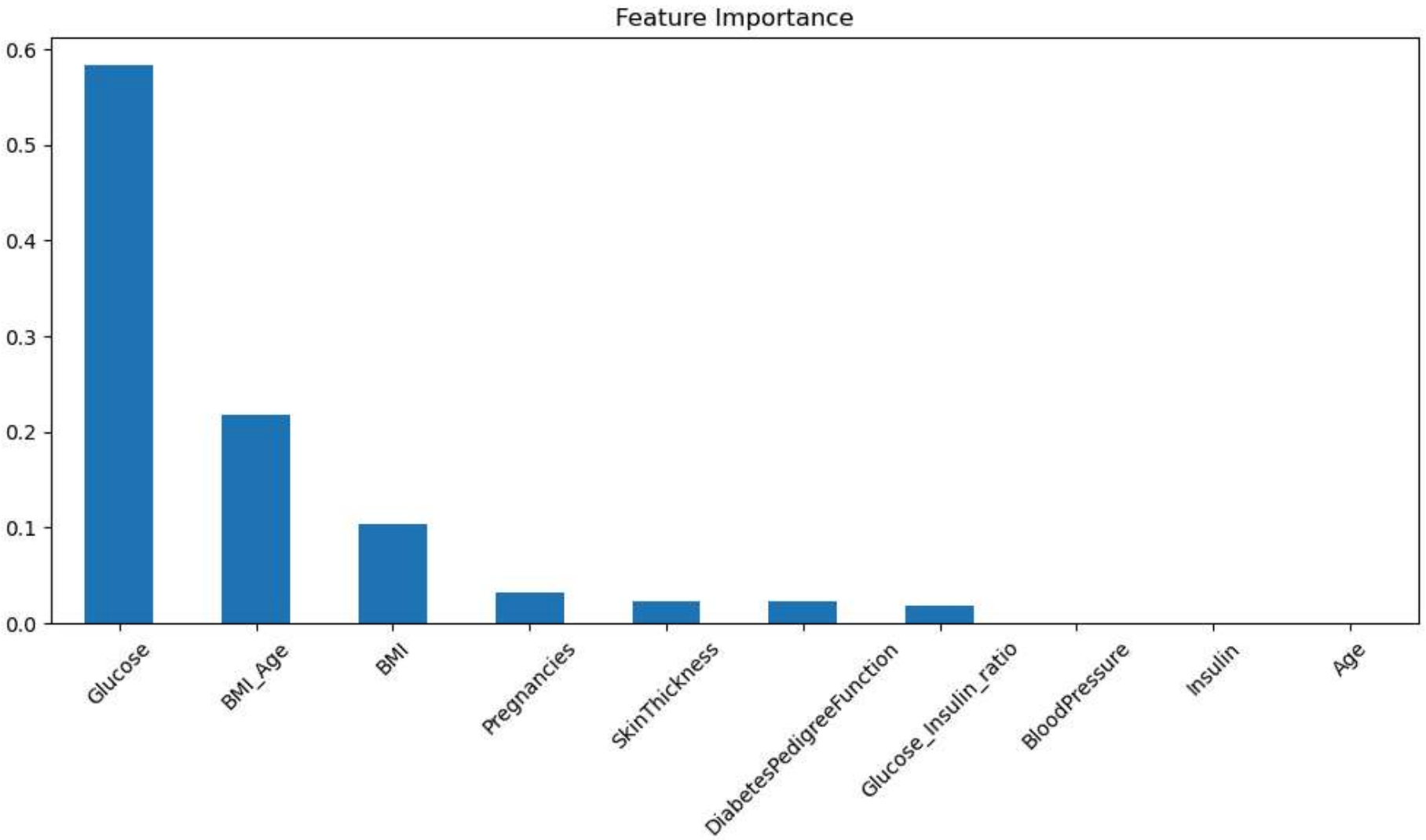
DecisionTreeClassifier

```
DecisionTreeClassifier(class_weight='balanced', max_depth=4,
                       min_samples_split=20, random_state=42)
```

In [6]:
```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc_auc_score
Y_pred = model.predict(X_test_scaled)
print("Accuracy",accuracy_score(Y_test, Y_pred))
print("ROC AUC:",roc_auc_score(Y_test,Y_pred))
print("\nClassification Report:\n", classification_report(Y_test, Y_pred))

print("\nConfusion Matrix:")
print(pd.crosstab(Y_test, Y_pred,
      rownames=['Actual'],
      colnames=['Predicted']))
```

```
Accuracy 0.7077922077922078
ROC AUC: 0.740925925925926

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.63      0.74       100
           1       0.55      0.85      0.67        54

    accuracy                           0.71       154
   macro avg       0.72      0.74      0.70       154
weighted avg       0.77      0.71      0.71       154


Confusion Matrix:
Predicted   0   1
Actual
0          63  37
1           8  46
```

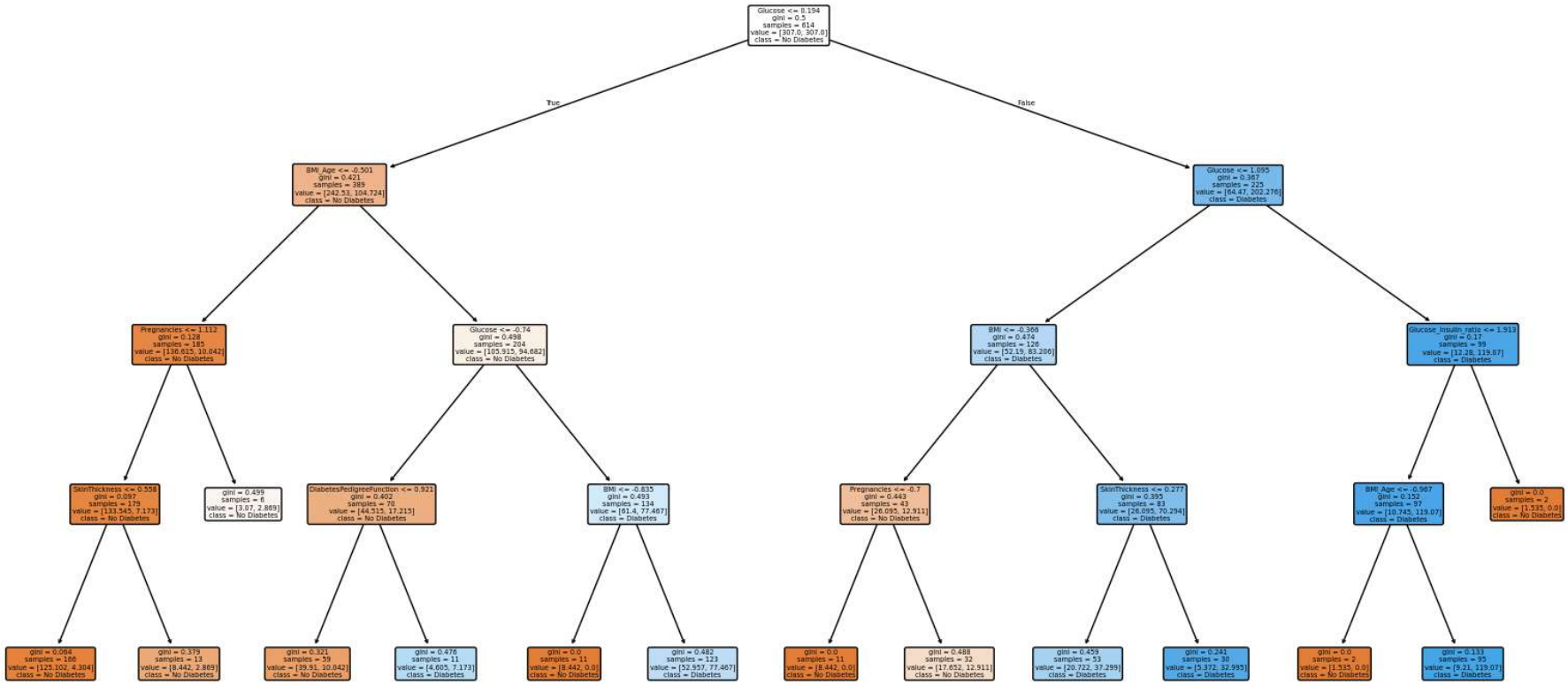
In [7]:
```python
import matplotlib.pyplot as plt
importances =pd.Series(model.feature_importances_, index= X.columns)
importances = importances.sort_values(ascending=False)

plt.figure(figsize=(10,6))
importances.plot(kind='bar')
plt.title("Feature Importance")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [8]:
```
from sklearn.model_selection import GridSearchCV

params = {
    'max_depth': [3, 5, 7],
    'min_samples_split': [10, 20, 30]
}
```

```
grid = GridSearchCV(DecisionTreeClassifier(), params, cv=5)
grid.fit(X_train_scaled, Y_train)
print("\nBest parameters:", grid.best_params_)

from sklearn.tree import plot_tree

plt.figure(figsize=(20,10))
plot_tree(model,
          feature_names=X.columns,
          class_names=['No Diabetes', 'Diabetes'],
          filled=True,
          rounded=True)
plt.show()
```

```
Best parameters: {'max_depth': 7, 'min_samples_split': 10}
```


Glucose <= 0.194
gini = 0.5
samples = 614
value = [307.0, 307.0]
class = No Diabetes
True
False
BMI_Age <= -0.501
gini = 0.421
samples = 389
value = [242.53, 104.724]
class = No Diabetes
Glucose <= 1.095
gini = 0.367
samples = 225
value = [64.47, 202.276]
class = Diabetes
Pregnancies <= 1.112
gini = 0.128
samples = 185
value = [136.615, 10.042]
class = No Diabetes
Glucose <= -0.74
gini = 0.498
samples = 204
value = [105.915, 94.682]
class = No Diabetes
BMI <= -0.366
gini = 0.474
samples = 126
value = [52.19, 83.206]
class = Diabetes
Glucose_Insulin_ratio <= 1.913
gini = 0.17
samples = 99
value = [12.28, 119.07]
class = Diabetes
SkinThickness <= 0.558
gini = 0.097
samples = 179
value = [133.545, 7.173]
class = No Diabetes
gini = 0.499
samples = 6
value = [3.07, 2.869]
class = No Diabetes
DiabetesPedigreeFunction <= 0.921
gini = 0.402
samples = 70
value = [44.515, 17.215]
class = No Diabetes
BMI <= -0.835
gini = 0.493
samples = 134
value = [61.4, 77.467]
class = Diabetes
Pregnancies <= -0.7
gini = 0.443
samples = 43
value = [26.095, 12.911]
class = No Diabetes
SkinThickness <= 0.277
gini = 0.395
samples = 83
value = [26.095, 70.294]
class = Diabetes
BMI_Age <= -0.967
gini = 0.152
samples = 97
value = [10.745, 119.07]
class = Diabetes
gini = 0.0
samples = 2
value = [1.535, 0.0]
class = No Diabetes
gini = 0.064
samples = 166
value = [125.102, 4.304]
class = No Diabetes
gini = 0.379
samples = 13
value = [8.442, 2.869]
class = No Diabetes
gini = 0.321
samples = 59
value = [39.91, 10.042]
class = No Diabetes
gini = 0.476
samples = 11
value = [4.605, 7.173]
class = Diabetes
gini = 0.0
samples = 11
value = [8.442, 0.0]
class = No Diabetes
gini = 0.482
samples = 123
value = [52.957, 77.467]
class = Diabetes
gini = 0.0
samples = 11
value = [8.442, 0.0]
class = No Diabetes
gini = 0.488
samples = 32
value = [17.652, 12.911]
class = No Diabetes
gini = 0.459
samples = 53
value = [20.722, 37.299]
class = Diabetes
gini = 0.241
samples = 30
value = [5.372, 32.995]
class = Diabetes
gini = 0.0
samples = 2
value = [1.535, 0.0]
class = No Diabetes
gini = 0.133
samples = 95
value = [9.21, 119.07]
class = Diabetes