In [119…

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,precision_score,f1_score,recall_score
from sklearn.model_selection import KFold, cross_val_score,cross_val_predict
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

In [87]:

```
df=pd.read_csv('C:/Users/NITISH BOKKA/Downloads/archive (10)/diabetes.csv')
```

In [89]:

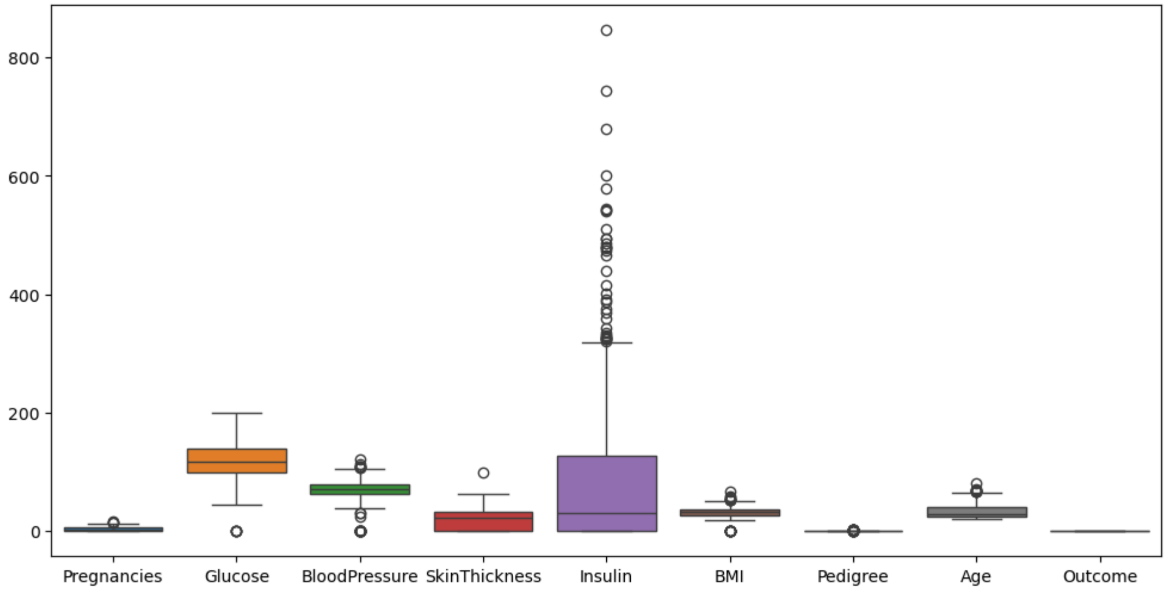
```
df.isnull().sum()
```

Out[89]:

```
Pregnancies      0
Glucose          0
BloodPressure    0
SkinThickness    0
Insulin          0
BMI              0
Pedigree         0
Age              0
Outcome          0
dtype: int64
```

In [91]:

```
plt.figure(figsize=(12,6))
sns.boxplot(df)
plt.show()
```

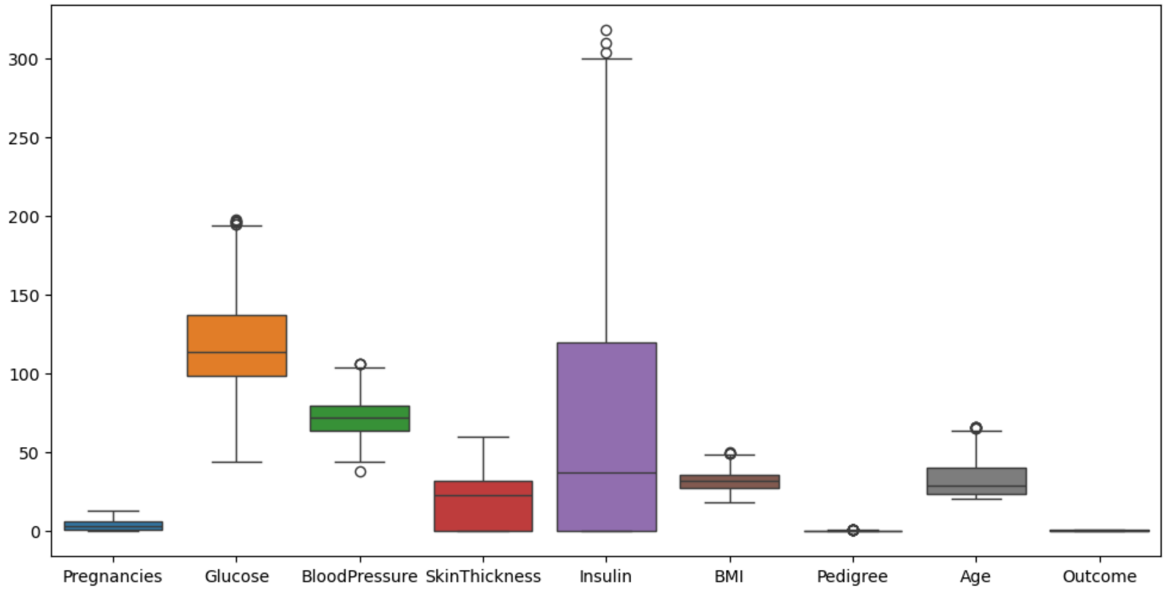


In [92]:

```
Q1=df.quantile(0.25)
Q3=df.quantile(0.75)
IQR=Q3-Q1
lower_quartile=Q1-1.5*IQR
upper_quartile=Q3+1.5*IQR
df=df[~((df<lower_quartile) | (df>upper_quartile)).any(axis=1)]
```

In [95]:

```
plt.figure(figsize=(12,6))
sns.boxplot(df)
plt.show()
```



In [97]:

```
X=df[['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','P
y=df['Outcome']
```

In [99]:

```
scaler=MinMaxScaler()
X_scaled=scaler.fit_transform(X)
```

In [101…

```
stScaler=StandardScaler()
X_scaler=stScaler.fit_transform(X_scaled)
```

In [103…

```
X_train,X_test,y_train,y_test=train_test_split(X_scaler,y,test_size=0.2,random_s
```

In [107…

```
neighbors=range(3,15)
for i in neighbors:
    knn=KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    y_pred=knn.predict(X_test)
    print(i, accuracy_score(y_test,y_pred))
```

```
3 0.7578125
4 0.7421875
5 0.8046875
6 0.7734375
7 0.8203125
8 0.7890625
9 0.8046875
10 0.7890625
11 0.8046875
12 0.765625
13 0.8046875
14 0.78125
```

In [121…

```
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train,y_train)
y_pred=knn.predict(X_test)
print('Accuracy', accuracy_score(y_test,y_pred))
print('Precision',precision_score(y_test,y_pred))
print('F1 Score',f1_score(y_test,y_pred))
print('Recall',recall_score(y_test,y_pred))
```

```
Accuracy 0.8203125
Precision 0.782608695652174
F1 Score 0.6101694915254238
Recall 0.5
```

In [123…

```
kf=KFold(n_splits=10,shuffle=True, random_state=42)
y_pred=cross_val_predict(knn,X,y,cv=kf)
accuracy=cross_val_score(knn,X,y,cv=kf,scoring='accuracy').mean()
print('Accuracy', accuracy)
print('Precision',precision_score(y,y_pred))
print('F1 Score',f1_score(y,y_pred))
print('Recall',recall_score(y,y_pred))
```

```
Accuracy 0.7464285714285714
Precision 0.6217948717948718
F1 Score 0.5449438202247191
Recall 0.485
```

In [ ]: