```
# Select the best model - RandomForest with manually tuned parameters
best_model = models['RandomForest']

# Evaluate the best model on the test set
y_pred_best = best_model.predict(X_test)

# Calculate performance metrics
accuracy_best = accuracy_score(y_test, y_pred_best)
precision_best = precision_score(y_test, y_pred_best)
recall_best = recall_score(y_test, y_pred_best)
f1_best = f1_score(y_test, y_pred_best)

# Display evaluation results
print(f"Accuracy: {accuracy_best:.2f}")
print(f"Precision: {precision_best:.2f}")
print(f"Recall: {recall_best:.2f}")
print(f"F1 Score: {f1_best:.2f}")

# Perform a sample prediction with the best performing model

# Select a random sample from the test set
sample = X_test.sample(1, random_state=42)

# Use the best model to predict the potability of the selected sample
sample_prediction = best_model.predict(sample)

# Retrieve the predicted result (0 = Not Potable, 1 = Potable)
sample_prediction_result = 'Potable' if sample_prediction[0] == 1 else 'Not Pot

# Display the selected sample and the prediction result
sample_data = sample.iloc[0].to_dict()
sample_data['Predicted Potability'] = sample_prediction_result

# Print sample features and prediction result
print("Sample Features:")
for key, value in sample_data.items():
    print(f"{key}: {value}")
"""
```