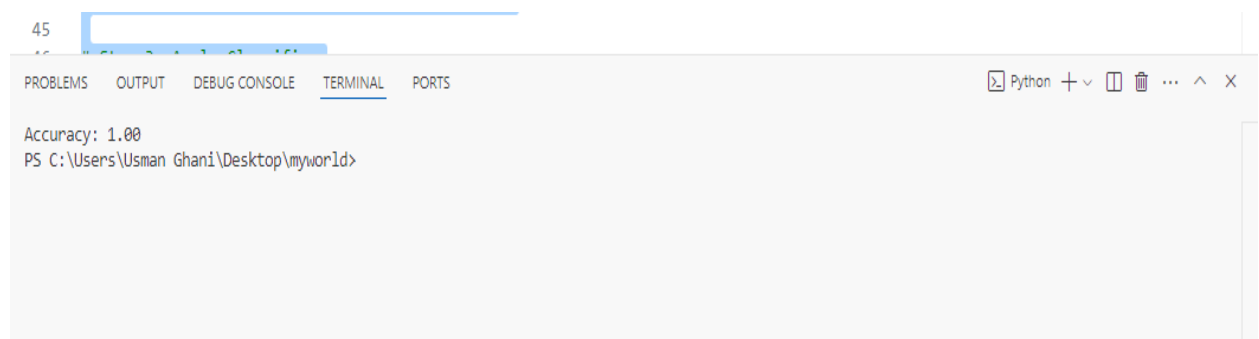


## AI\_LAB\_Task10

- Import required libraries from scikit-learn:
  - `train_test_split` for splitting the dataset.
  - `RandomForestClassifier` for building the classification model.
  - `accuracy_score` for evaluating the model's performance.
- Load the Iris dataset using pandas (`pd.read_csv("Iris.csv")`), though it's not used later in the code.
- Load the built-in Iris dataset using scikit-learn's `load_iris()` function.
- Extract features (X) and labels (y) from the dataset.
- Split the dataset into training and testing sets using `train_test_split`:
  - 80% of data used for training.
  - 20% of data used for testing.
  - `random_state` ensures reproducibility.
- Initialize the Random Forest Classifier with a fixed `random_state` for consistent results.
- Train the model on the training data using `model.fit()`.
- Predict the labels for the test set using `model.predict()`.
- Calculate the accuracy of the model by comparing predicted and actual labels using `accuracy_score`.
- Print the accuracy score rounded to two decimal places.

OUTPUT:



The screenshot shows a Python IDE interface with a terminal window. The terminal displays the output of a RandomForestClassifier model. The accuracy is 1.00, and the command prompt is PS C:\Users\Usman Ghani\Desktop\myworld>.

```
45  
Accuracy: 1.00  
PS C:\Users\Usman Ghani\Desktop\myworld>
```