

✓ Mohammad Shayaan Shaikh (54)

EXP.NO.12

✓ Program to demonstrate DataFrame using Pandas

```
import pandas as pd
```

✓ Creating a DataFrame using a dictionary

```
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [24, 27, 22, 32],
    'Marks': [85, 78, 90, 88]
}
```

```
df = pd.DataFrame(data)
print("Initial DataFrame:\n", df)
```

```
↗ Initial DataFrame:
```

	Name	Age	Marks
0	Alice	24	85
1	Bob	27	78
2	Charlie	22	90
3	David	32	88

✓ Accessing columns

```
print("Accessing 'Name' column:\n", df['Name'])
```

```
↗ Accessing 'Name' column:
```

0	Alice
1	Bob
2	Charlie
3	David

Name: Name, dtype: object

✓ Adding a new column

```
df['Grade'] = ['B', 'C', 'A', 'A']
print("DataFrame after adding 'Grade' column:\n", df)
```

```
↗ DataFrame after adding 'Grade' column:
```

	Name	Age	Marks	Grade
0	Alice	24	85	B
1	Bob	27	78	C
2	Charlie	22	90	A
3	David	32	88	A

✓ Accessing a specific row using loc

```
print("Accessing row at index 2:\n", df.loc[2])
```

```
↗ Accessing row at index 2:
```

Name	Charlie
Age	22
Marks	90
Grade	A

Name: 2, dtype: object

✓ Accessing multiple rows using iloc

```
print("Accessing rows from index 1 to 3:\n", df.iloc[1:4])
```

```
↗ Accessing rows from index 1 to 3:
```

	Name	Age	Marks	Grade
1	Bob	27	78	C
2	Charlie	22	90	A
3	David	32	88	A

▼ Filtering data

```
high_scorers = df[df['Marks'] > 85]
print("Students scoring above 85 marks:\n", high_scorers)
```

↗ Students scoring above 85 marks:

	Name	Age	Marks	Grade
2	Charlie	22	90	A
3	David	32	88	A

▼ Basic statistics

```
print("Statistical summary:\n", df.describe())
```

↗ Statistical summary:

	Age	Marks
count	4.000000	4.000000
mean	26.250000	85.250000
std	4.349329	5.251984
min	22.000000	78.000000
25%	23.500000	83.250000
50%	25.500000	86.500000
75%	28.250000	88.500000
max	32.000000	90.000000