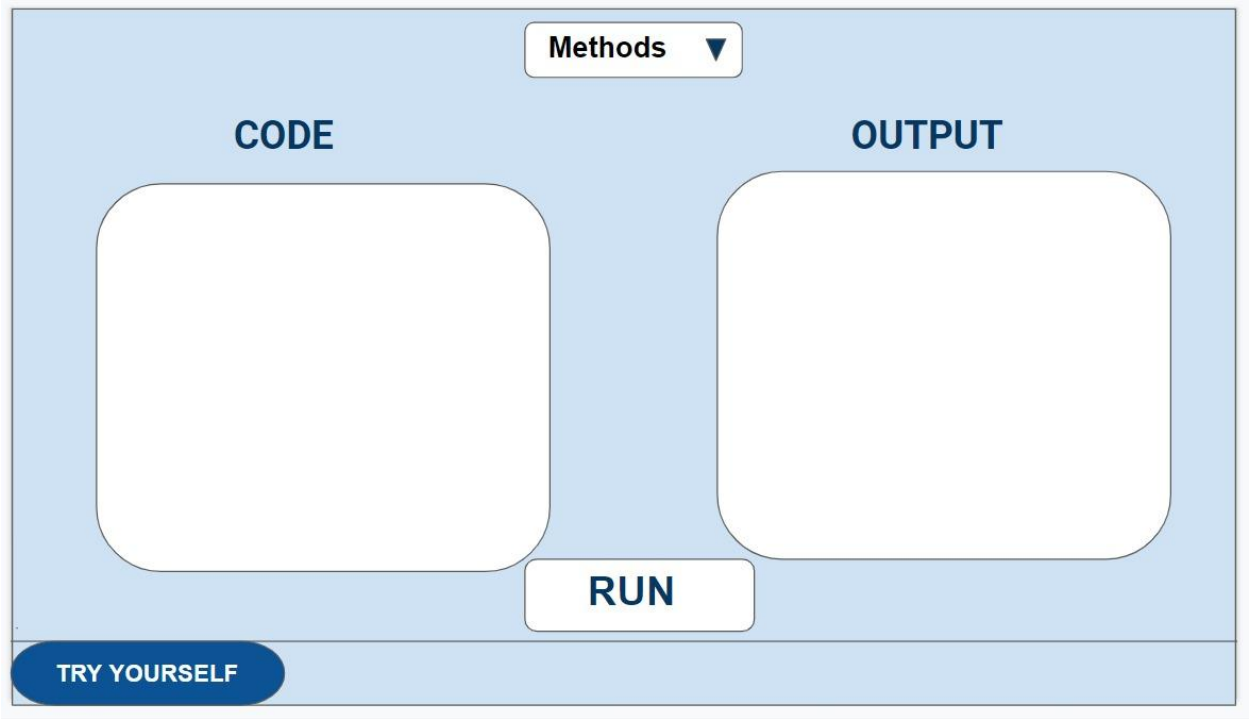
Virtual Labs Experiment (Simulation)

Adding new rows in a dataframe



Layout of the experiment (Simulation)

STEPS -

Step 1: We have created a DataFrame that we'll use throughout the experiment.

Step2: Various methods are available in Python to add rows in any data frame. Hence we'll provide the list of the various techniques and their codes.

Step3: The user selects the method and runs the respective code.

Layout Description:

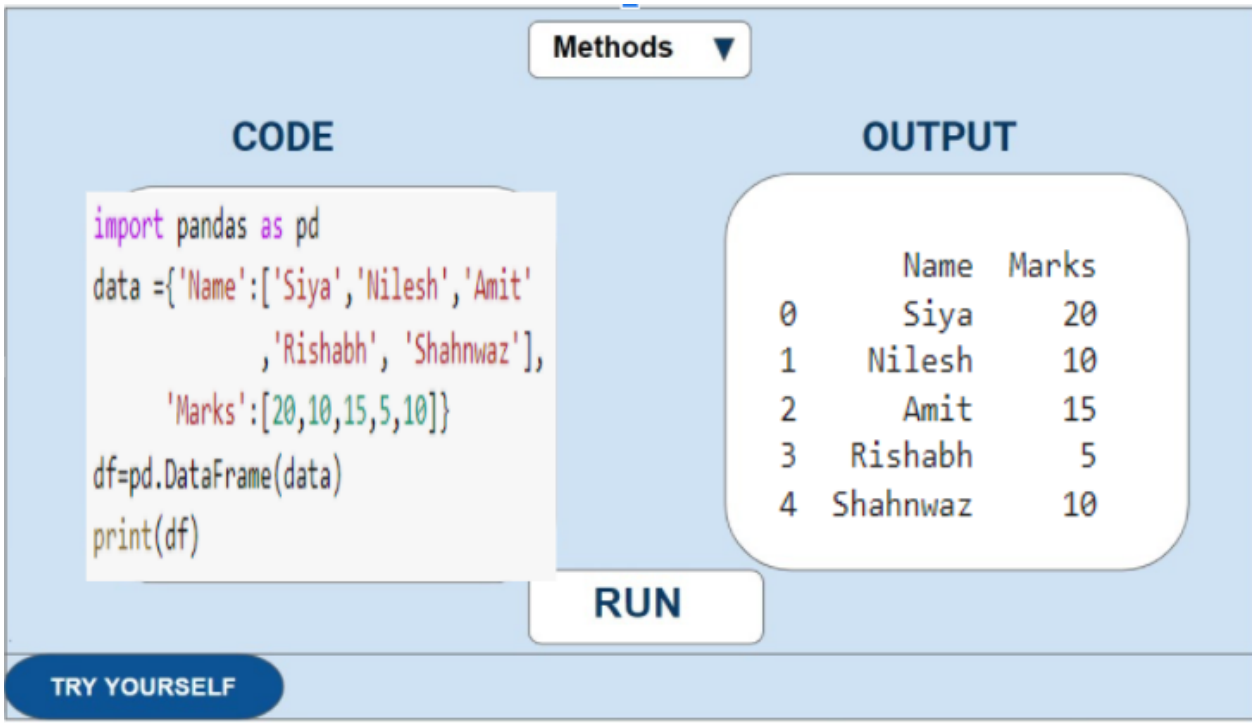
- A code section is provided on the left side, where all the sample codes will be displayed.
- A run button is provided at the bottom to execute the code..
- The output section is given on the right side, where all the generated outputs will be displayed.
- A "Try Yourself" button is provided to the users to try their own codes.

- Three methods are provided at the top of the layout in a dropdown menu, which can be accessed through the "Method" button.

Steps to follow(Simulation Content):

Here we have demonstrated three different methods on "adding rows in DataFrame" and one code to create the DataFrame that will be used throughout the simulation..

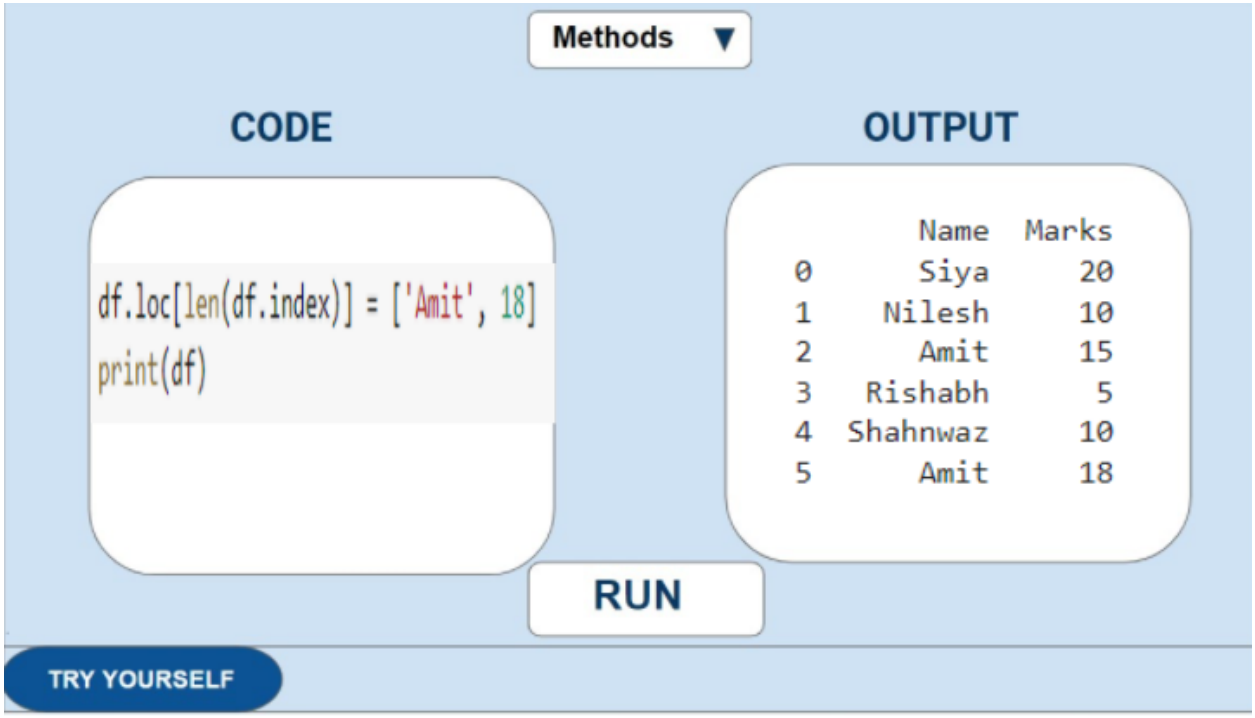
1. This is the very first page of the simulation. The code for creating a dataframe. Sample code is provided with the run button beneath it. On pressing the run button, the output of the code is shown.
   Button "TRY YOURSELF" is provided to the users to try their own codes.



code//-

```
import pandas as pd
data ={'Name':['Siya','Nilesh','Amit'
                ,'Rishabh', 'Shahnawaz'],
       'Marks':[20,10,15,5,10]}
df=pd.DataFrame(data)
print(df)
```

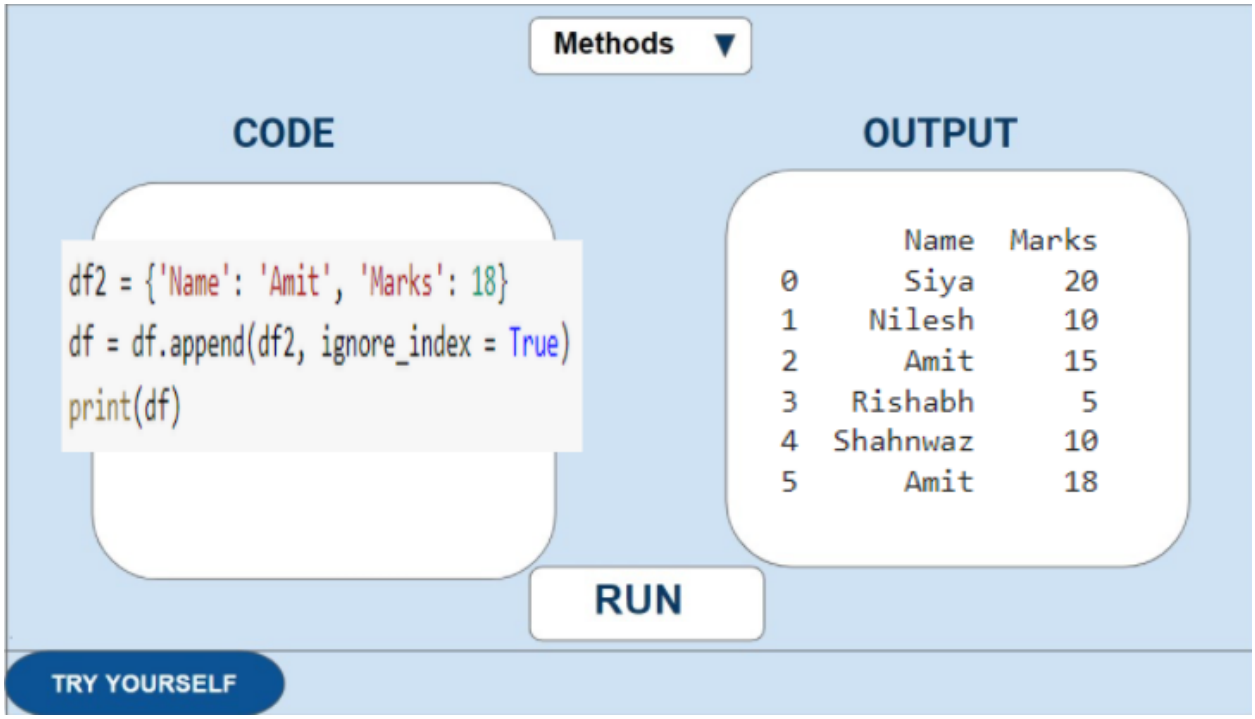
1) This is the Page2/Method 1 i.e Using Python **loc()** method to add row in data frame .Right box shows the updated content in the data frame as the output..



code//-

```
df.loc[len(df.index)] = ['Amit', 18]
print(df)
```

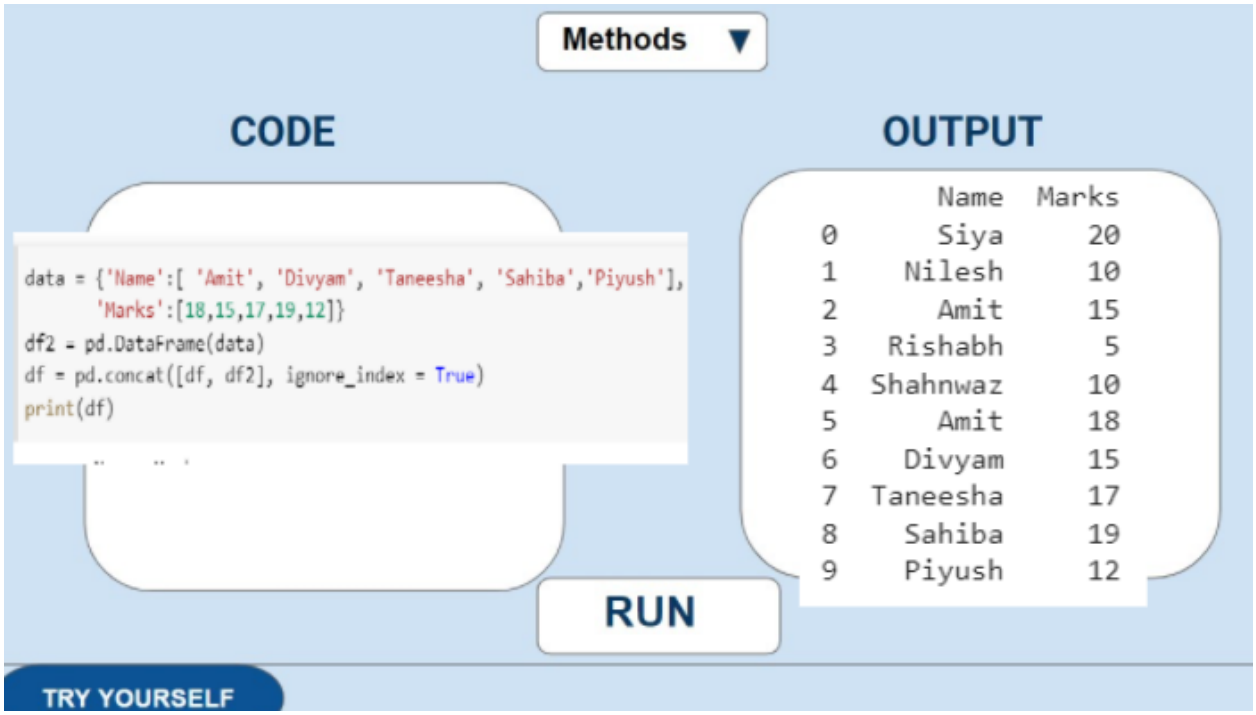
2) This is the Page3/Method 2 i.e Python **append()** method



code//-

```
df2 = {'Name': 'Amit', 'Marks': 18}
df = df.append(df2, ignore_index = True)
print(df)
```

4) This is the Page4/Method 3 i.e Python **pandas.concat()** method to concatenate two dataframes having the same number of columns.



code//-

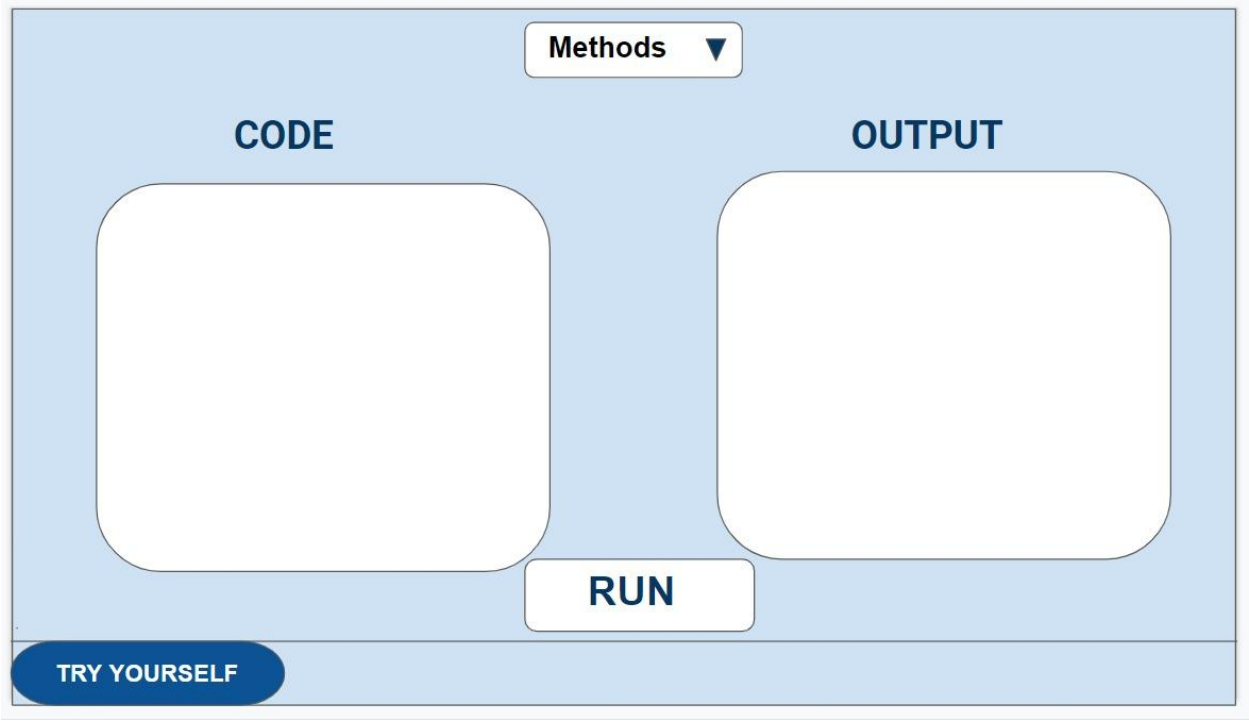
```
data = {'Name':['Amit','Divyam','Taneesha','Sahiba','Piyush'],
        'Marks':[18,15,17,19,12]}
df2 = pd.DataFrame(data)
df = pd.concat([df, df2], ignore_index = True)
print(df)
```

//

Content

Compute properties of a data frame and do manipulate its content (add rows)

Layout:



Theory:

What is Adding Rows in DataFrame:

The addition of content in the data frame is referred to as adding rows in a dataframe . There are many ways to add rows in a data frame out of which we detail three different methods below::

Sample DataFrame:

```
import pandas as pd
info= {"Num":[12,14,13,12,14,13,15],
"NAME":['John','Camili','Rheana','Joseph','Amanti','Alexa','Siri
']}

data = pd.DataFrame(info)
print("Original Data frame:\n")
```

```
print(data)
```

1. Using Python loc() method to add a single row in the dataframe.

Python loc() method enables us to add one row at the specified location of the data frame at a time. In order to add a row at the end of a data frame, **len(dataFrame.index)** is used which gives the total number of rows in the data frame.

Syntax:

dataFrame.loc[len(dataFramef.index)]=' values '

Example:

dataFrame.loc[len(dataFrame.index)] = [18 , 'Amit']

2. Python append() function to add a row.

Python append() method may be used to append a new row in a data frame using data dictionary or another data frame where append means adding row at the end of the dataframe.

Syntax:

dataframe.append(another_dataframe/datadictionary , ignore_index = True)

Example:

```
df2 = {'Num': '18' , 'Name': 'amit'}

data = df.append(df2, ignore_index = True)
print(data)
```

3. Python concat() method to add rows of a data frame into another data frame . This method concatenates two or more dataframes into one .

Syntax:

pandas.concat([dataframe1, dataframe2], ignore_index = True)

Example

```
data={Num':[18,15,17,19,12],'Name':['Amit','Divyam',
'Taneesha','Sahiba','Piyush']'}
df2 = pd.DataFrame(data)
df = pd.concat([data, df2], ignore_index = True)
print(df)
```

Procedure:

- On the first page, a DataFrame is provided that will be used throughout the experiment.
- A code section is provided on the left side, where all the sample codes are displayed.
- A run button is provided at the bottom.
- The output section is given on the right side, where all the generated outputs are displayed.
- Various methods are available in Python to add rows in the dataframe. Hence a list of the different techniques and their codes is provided in a dropdown menu, which can be accessed through the "Method" button.
- **S**elect the desired method and run the respective code.
- A "Try Yourself" button is also provided to try other code combinations.

Simulation :

Following steps are to be shown:

1. **Creating DataFrame:**

```
import pandas as pd
data ={'Name':['Siya','Nilesh','Amit'
               ,'Rishabh', 'Shahnawaz'],
      'Marks':[20,10,15,5,10]}
df=pd.DataFrame(data)
print(df)
```

2. **Using Python loc() method to add rows in dataframe..**

```
df.loc[len(df.index)] = ['Amit', 18]
print(df)
```

3. **Using Python append() method to add rows in dataframe..**

```
df2 = {'Name': 'Amit', 'Marks': 18}
df = df.append(df2, ignore_index = True)
print(df)
```

4. **Using Python concat() method to add rows in dataframe..**

```
data={'Name':['Amit','Divyam','Taneesha','Sahiba','Piyush'],
      'Marks':[18,15,17,19,12]}
df2 = pd.DataFrame(data)
df = pd.concat([df, df2], ignore_index = True)
print(df)
```