Fenway's CorePower Yoga

Milestone: Application

Group 19

Venkata Nithya Ala

Sri Nikitha Reddy Karedla

617-820-9583

857-313-4062

ala.v@northeastern.edu

karedla.sr@northeastern.edu

Percentage of Effort Contributed by Student1: 50

Percentage of Effort Contributed by Student2: 50

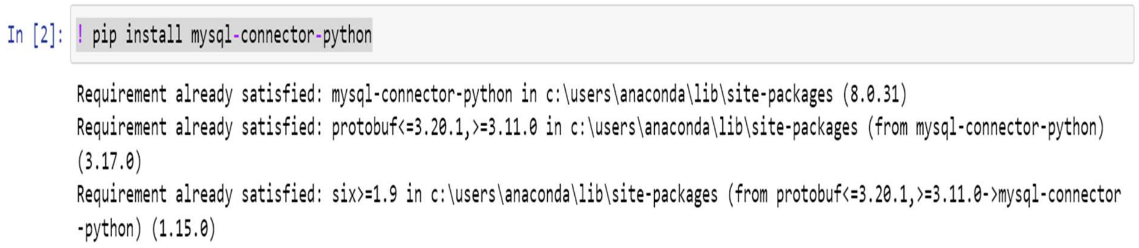
Signature of Student 1: Venkata Nithya Ala

Signature of Student 2: Sri Nikitha Reddy Karedla

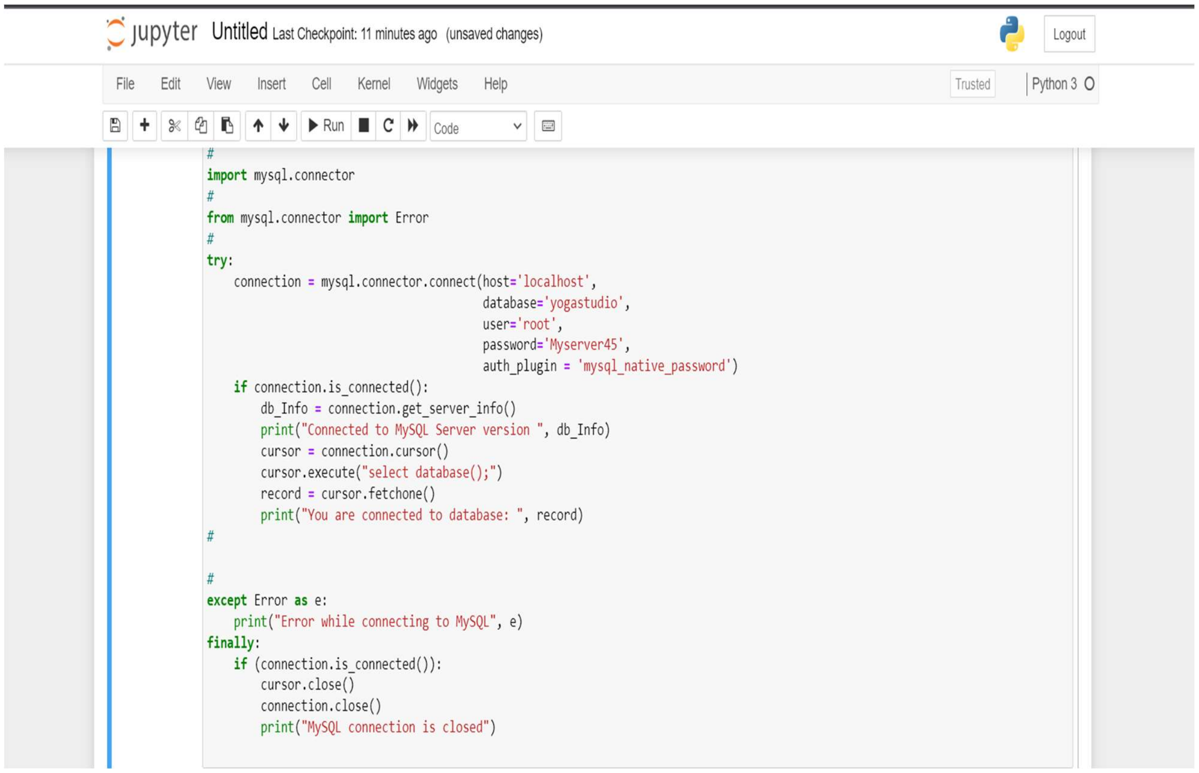
Submission Date: November 26, 2022

Progress Report:

- Connected to MySQL database ‘yogastudio’ using python.



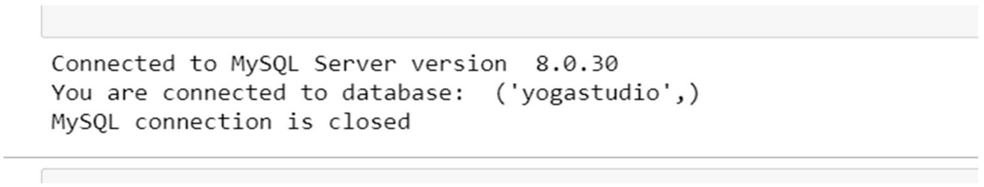
```
In [2]: ! pip install mysql-connector-python

Requirement already satisfied: mysql-connector-python in c:\users\anaconda\lib\site-packages (8.0.31)
Requirement already satisfied: protobuf<=3.20.1,>=3.11.0 in c:\users\anaconda\lib\site-packages (from mysql-connector-python)
(3.17.0)
Requirement already satisfied: six>=1.9 in c:\users\anaconda\lib\site-packages (from protobuf<=3.20.1,>=3.11.0->mysql-connector
-python) (1.15.0)
```



jupyter Untitled Last Checkpoint: 11 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

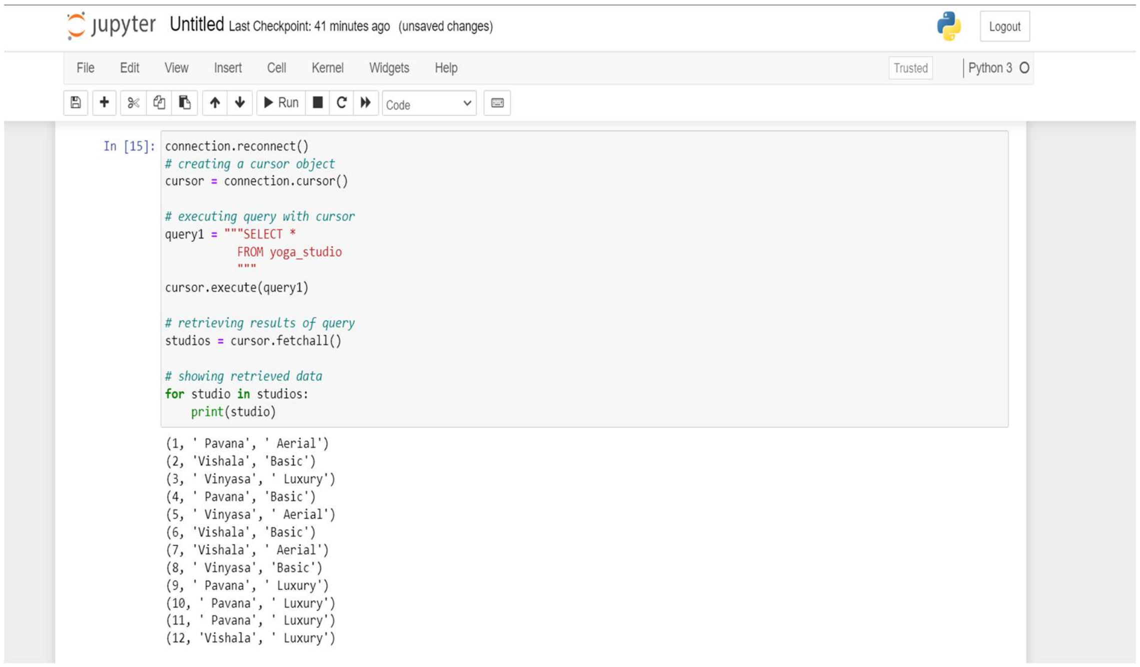
```python
#
import mysql.connector
#
from mysql.connector import Error
#
try:
    connection = mysql.connector.connect(host='localhost',
                                         database='yogastudio',
                                         user='root',
                                         password='Myserver45',
                                         auth_plugin = 'mysql_native_password')
    if connection.is_connected():
        db_Info = connection.get_server_info()
        print("Connected to MySQL Server version ", db_Info)
        cursor = connection.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You are connected to database: ", record)
#

#
except Error as e:
    print("Error while connecting to MySQL", e)
finally:
    if (connection.is_connected()):
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

Output:



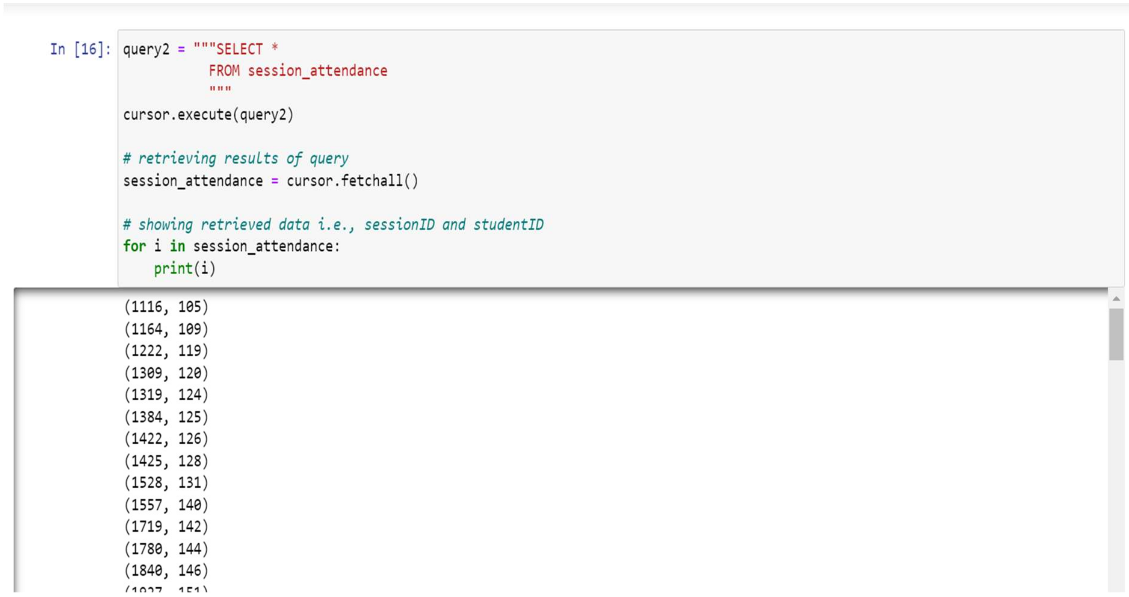
```
Connected to MySQL Server version  8.0.30
You are connected to database:  ('yogastudio',)
MySQL connection is closed
```

- Querying the database using python:


jupyter Untitled Last Checkpoint: 41 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

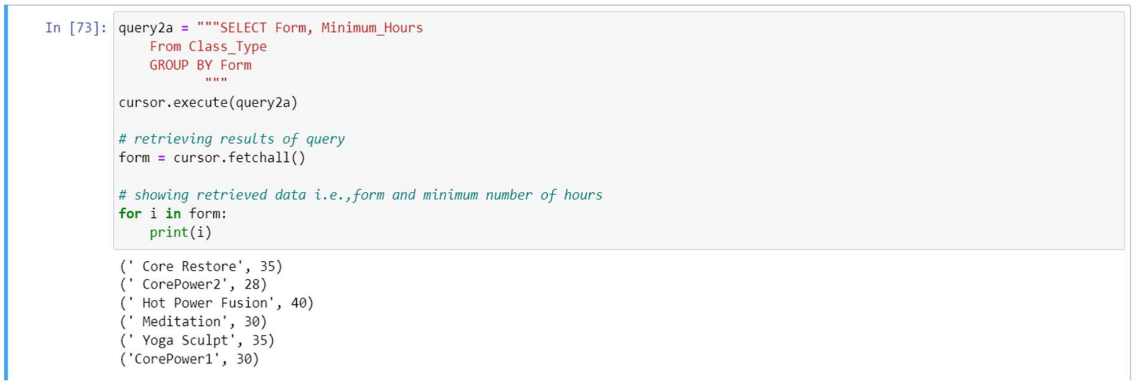
In [15]:
```
connection.reconnect()
# creating a cursor object
cursor = connection.cursor()

# executing query with cursor
query1 = """SELECT *
           FROM yoga_studio
           """
cursor.execute(query1)

# retrieving results of query
studios = cursor.fetchall()

# showing retrieved data
for studio in studios:
    print(studio)
```

```
(1, ' Pavana', ' Aerial')
(2, 'Vishala', 'Basic')
(3, ' Vinyasa', ' Luxury')
(4, ' Pavana', 'Basic')
(5, ' Vinyasa', ' Aerial')
(6, 'Vishala', 'Basic')
(7, 'Vishala', ' Aerial')
(8, ' Vinyasa', 'Basic')
(9, ' Pavana', ' Luxury')
(10, ' Pavana', ' Luxury')
(11, ' Pavana', ' Luxury')
(12, 'Vishala', ' Luxury')
```



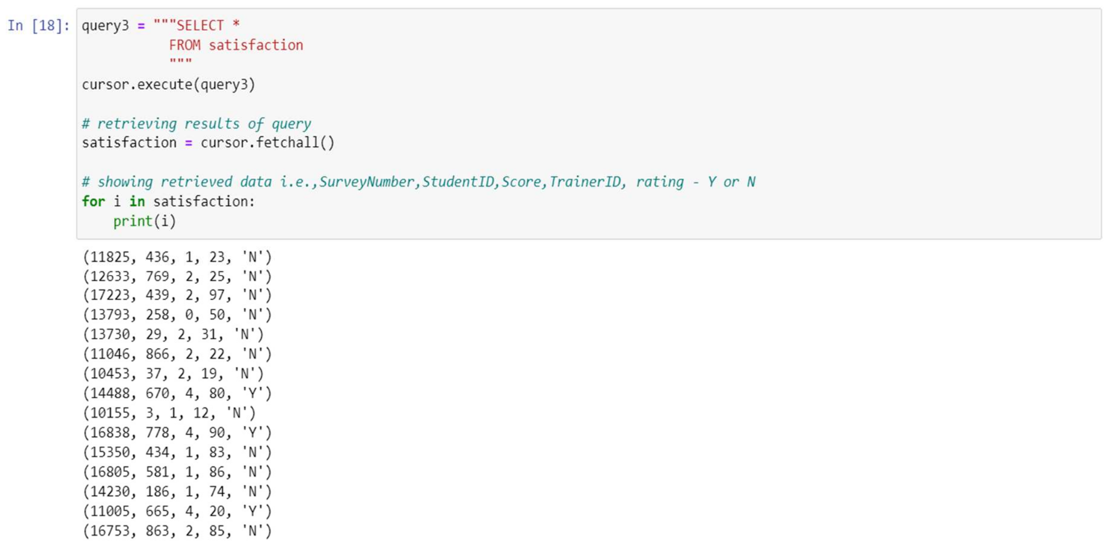
In [16]:
```
query2 = """SELECT *
           FROM session_attendance
           """
cursor.execute(query2)

# retrieving results of query
session_attendance = cursor.fetchall()

# showing retrieved data i.e., sessionID and studentID
for i in session_attendance:
    print(i)
```

```
(1116, 105)
(1164, 109)
(1222, 119)
(1309, 120)
(1319, 124)
(1384, 125)
(1422, 126)
(1425, 128)
(1528, 131)
(1557, 140)
(1719, 142)
(1780, 144)
(1840, 146)
```

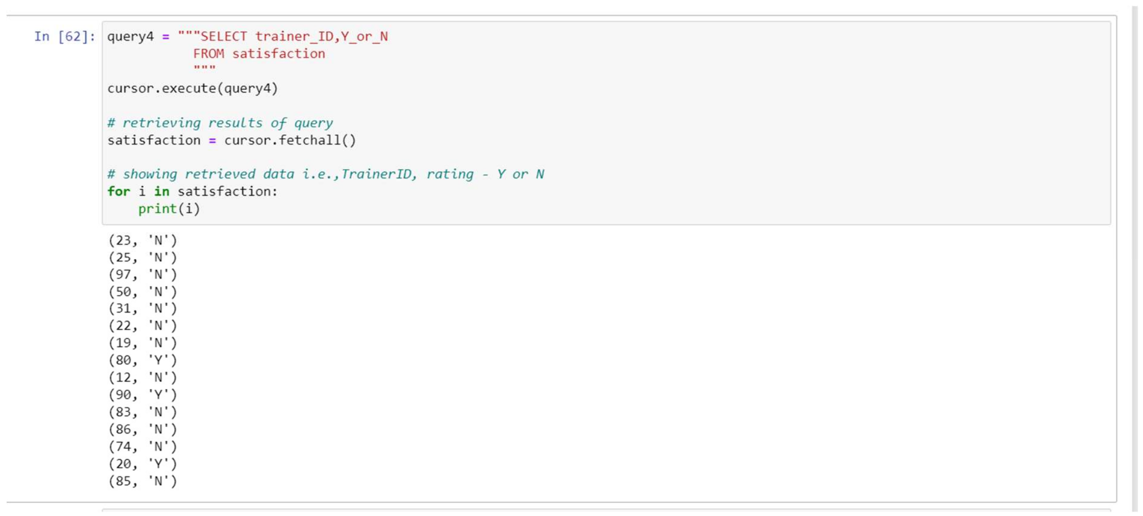


In [73]:
```
query2a = """SELECT Form, Minimum_Hours
    From Class_Type
    GROUP BY Form
           """
cursor.execute(query2a)

# retrieving results of query
form = cursor.fetchall()

# showing retrieved data i.e.,form and minimum number of hours
for i in form:
    print(i)
```

```
(' Core Restore', 35)
(' CorePower2', 28)
(' Hot Power Fusion', 40)
(' Meditation', 30)
(' Yoga Sculpt', 35)
('CorePower1', 30)
```



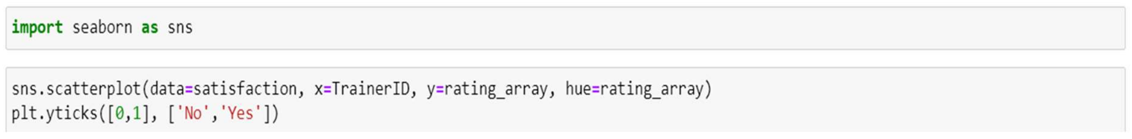
```
In [18]: query3 = """SELECT *
                    FROM satisfaction
                    """
         cursor.execute(query3)

         # retrieving results of query
         satisfaction = cursor.fetchall()

         # showing retrieved data i.e.,SurveyNumber,StudentID,Score,TrainerID, rating - Y or N
         for i in satisfaction:
             print(i)

         (11825, 436, 1, 23, 'N')
         (12633, 769, 2, 25, 'N')
         (17223, 439, 2, 97, 'N')
         (13793, 258, 0, 50, 'N')
         (13730, 29, 2, 31, 'N')
         (11046, 866, 2, 22, 'N')
         (10453, 37, 2, 19, 'N')
         (14488, 670, 4, 80, 'Y')
         (10155, 3, 1, 12, 'N')
         (16838, 778, 4, 90, 'Y')
         (15350, 434, 1, 83, 'N')
         (16805, 581, 1, 86, 'N')
         (14230, 186, 1, 74, 'N')
         (11005, 665, 4, 20, 'Y')
         (16753, 863, 2, 85, 'N')
```



```
In [62]: query4 = """SELECT trainer_ID,Y_or_N
                    FROM satisfaction
                    """
         cursor.execute(query4)

         # retrieving results of query
         satisfaction = cursor.fetchall()

         # showing retrieved data i.e.,TrainerID, rating - Y or N
         for i in satisfaction:
             print(i)

         (23, 'N')
         (25, 'N')
         (97, 'N')
         (50, 'N')
         (31, 'N')
         (22, 'N')
         (19, 'N')
         (80, 'Y')
         (12, 'N')
         (90, 'Y')
         (83, 'N')
         (86, 'N')
         (74, 'N')
         (20, 'Y')
         (85, 'N')
```

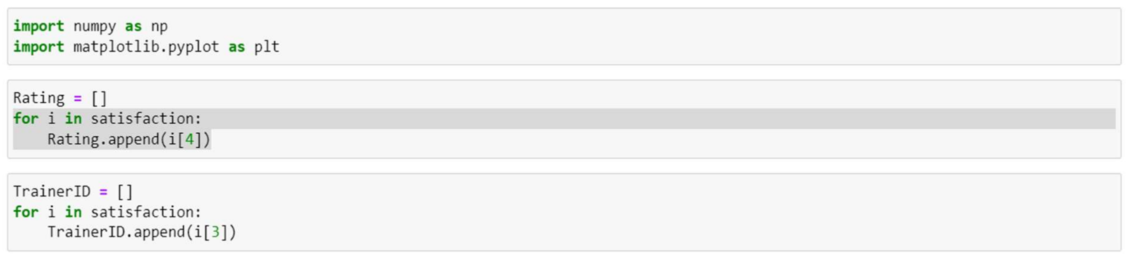
- Visualization of data using python:

  Plotting TrainerID vs Rating to know how many trainers are performing well:



```
import seaborn as sns
```

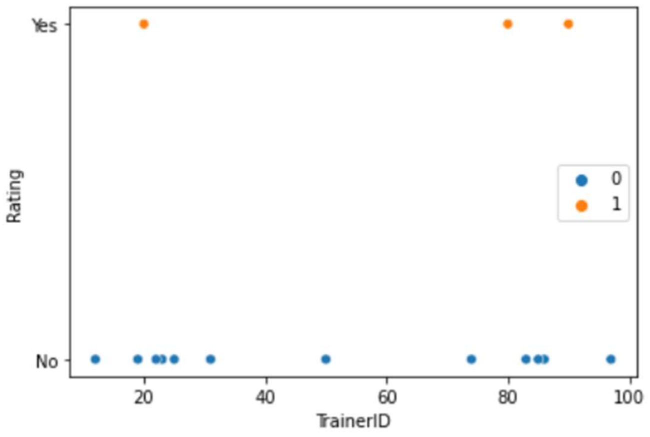
```
sns.scatterplot(data=satisfaction, x=TrainerID, y=rating_array, hue=rating_array)
plt.yticks([0,1], ['No','Yes'])
```



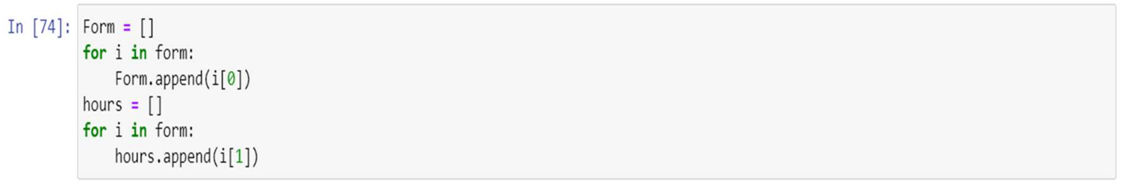
```
import numpy as np
import matplotlib.pyplot as plt
```

```
Rating = []
for i in satisfaction:
    Rating.append(i[4])
```

```
TrainerID = []
for i in satisfaction:
    TrainerID.append(i[3])
```

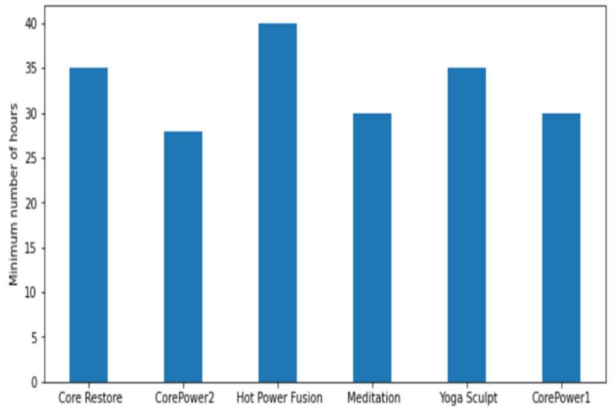


Plotting the minimum number of hours required to learn a yoga form:



```
In [74]: Form = []
         for i in form:
             Form.append(i[0])
         hours = []
         for i in form:
             hours.append(i[1])
```

```
In [81]: fig = plt.figure(figsize = (10, 5))
         plt.bar(Form,hours, width = 0.4)
         plt.ylabel('Minimum number of hours')

Out[81]: Text(0, 0.5, 'Minimum number of hours')
```


Minimum number of hours
Core Restore
CorePower2
Hot Power Fusion
Meditation
Yoga Sculpt
CorePower1