22p-9318_Ahmed_BS(SE)-5B_LAB-TASK-04_Q1

September 28, 2024

[9]:

```
import re
date_pattern=r"^(0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(1[0-9]{3}|2[0-9]{3})$"
#day part accepts values from 01 to 31 the pattern 0[1-9]|[12][0-9]|3[01]␣
 ↪ensures it
#month part accepts values from 01 to 12 the pattern 0[1-9]|1[0-2] ensures it
#year part accepts any 4 digit number starting from 1000 to 9999 the pattern␣
 ↪1[0-9]{3}|2[0-9]{3} ensures it
```

[10]:

```
date_input=input("Enter date in the format DD/MM/YYYY: ")
```

```
Enter date in the format DD/MM/YYYY:  29/02/2024
```

[13]:

```
check=re.findall(date_pattern, date_input)
```

[14]:

```
if check!=[]:
    print("valid match, moving to validate day month and year")
    day=int(check[0][0]) #converting days from string into an integer
    month=int(check[0][1]) #converting months from string into an integer
    year=int(check[0][2]) #converting years from string into an integer
else:
    print("invalid date: The format or values are incorrect")
```

```
valid match, moving to validate day month and year
```

[15]:

```
if check!=[]:
    if month==2: #special case for February
        if(year%4==0 and year%100!=0) or (year%400==0):
            #leap year check for February(can have 29 days in a leap year)
            if day>29:
                print("invalid date: February can't have more than 29 days in a␣
 ↪leap year")
            else:
                print("valid date: February in a leap year")
        else:
            #not a leap year check for February(can only have 28 days)
            if day>28:
                print("invalid date: February can't have more than 28 days in a␣
 ↪non leap year")
```

```
            else:
                print("valid date: February in a non leap year")
    elif month in [4, 6, 9, 11]: #months with 30 days(April, June, September,␣
↪November)
        if day>30:
            print(f"invalid date: Month {month} can't have more than 30 days.")
        else:
            print("valid date for a month with 30 days")
    else: #For all other months with 31 days

        print("valid date for a month with 31 days")
```

```
valid date: February in a leap year
```

[16]:

```
if check!=[]:
    print(f"Date entered: {date_input}")
    print(f"Day: {day}, Month: {month}, Year: {year}")
```

```
Date entered: 29/02/2024
Day: 29, Month: 2, Year: 2024
```

[ ]:

```
```