RCOEM

Shri Ramdeobaba College of Engineering and Management, Nagpur

Software Engineering T.A. 02

REPORT ON :

Software Testing on TodoList

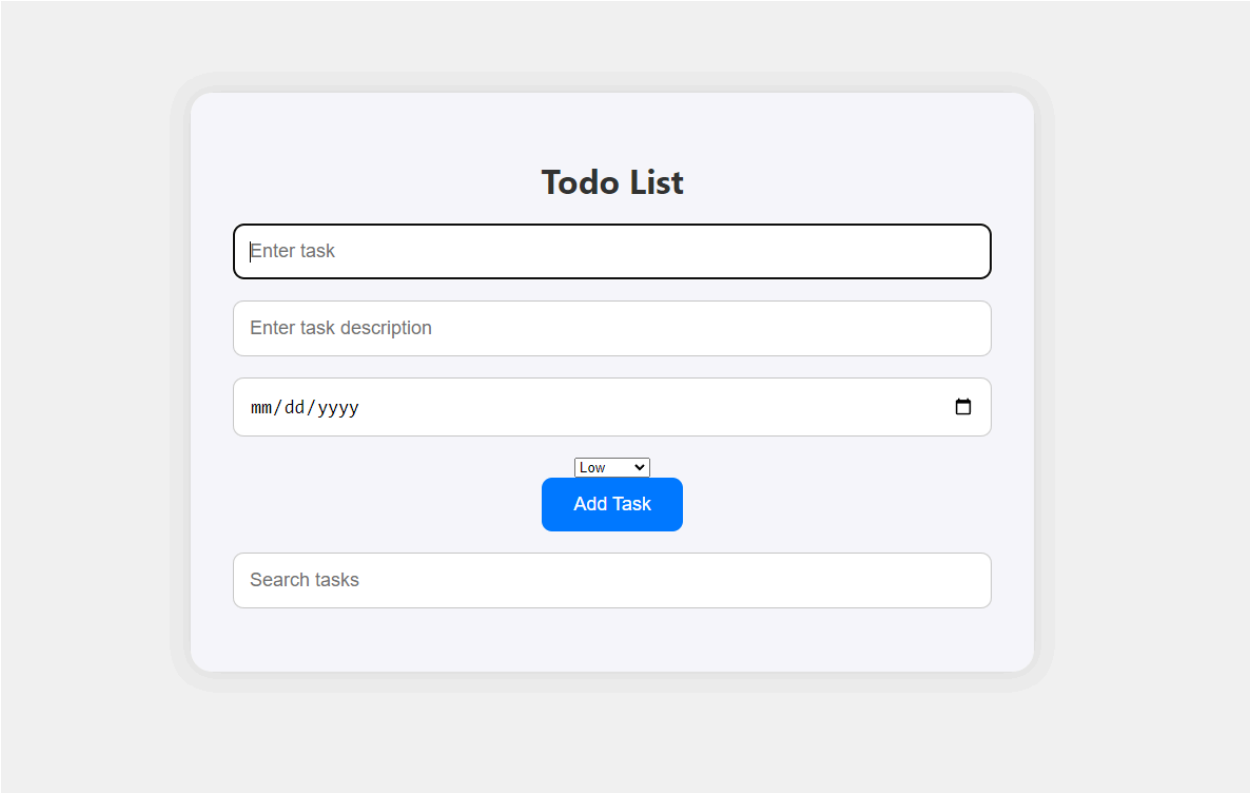
Created By -

Name : Kunal Dewangan

Roll No. : 41

Black Box Testing

TodoList -



Test Cases	Expected Output	Actual Output	Remark
Enter Task Title	Task Title should be added properly	Task Title added successfully	Positive
Enter Task Description	Task Description Should be added properly	Task Description added successfully	Positive
Add Due Date	There should be a calendar in which we can add due date	Due Date added successfully	Positive
Add priority level	There should be options for priority level which we can select	There are 3 options with low, medium and high. Priority level added successfully	Positive
Working of Done	After the Task is completed, there should be an option to done the task	After Clicking on done button, the task block turns green	Positive
Double click on done	After the task block turns green if we again click on the done button , no changes should occur	I clicked on the done button again , the task block become white which shows that the task being undone	Negative
Updation of Task Title And Description	After clicking on update button , task title and description should be updated properly	Task Title and Description updated successfully	Positive
Updation of Due date and priority	There should be option for updating Due date and Priority	There is no option to update due date and priority	Negative
Delete Button	The delete button should work after the task is completed	Delete button is working even before task is completed	Negative

Todo List

Enter task

Enter task description

mm/dd/yyyy

Low

Add Task

Search tasks

X

Task1

5 Questions of DSA
Due Date: 2024-04-12
Priority: high

Done Update

X

Task2

Deep Learning Project
Due Date: 2024-04-11
Priority: high

Done Update

Todo List

Enter task

Enter task description

mm/dd/yyyy

Low

Add Task

Search tasks

Task1

5 Questions of DSA
Due Date: 2024-04-12
Priority: high

Done Update

X

White Box Testing

Calculator(Backend) -

```
import math

class Calculator:
    def add(self, a, b):
        return a + b

    def subtract(self, a, b):
        return a - b

    def multiply(self, a, b):
        return a * b

    def divide(self, a, b):
        if b == 0:
            print("Cannot divide by zero")
            return
        return a / b

class ScientificCalculator(Calculator):
    def power(self, a, b):
        return a ** b

    def square_root(self, a):
        if a < 0:
            print("Cannot find square root of a negative number")
            return
        return a ** 0.5
```

```
    def natural_logarithm(self, a):
        if a <= 0:
            print("Input must be greater than 0 for natural logarithm")
            return
        return math.log(a)

    def base_10_logarithm(self, a):
        if a <= 0:
            print("Input must be greater than 0 for base-10 logarithm")
            return
        return math.log10(a)

    def factorial(self, n):
        return math.factorial(n)

class TrigonometricCalculator(Calculator):
    import math

    def sin(self, angle):
        return self.math.sin(angle)

    def cos(self, angle):
        return self.math.cos(angle)

    def tan(self, angle):
        return self.math.tan(angle)

    def inverse_sin(self, x):
        return math.asin(x)
```

```
def inverse_cos(self, x):
    return math.acos(x)

def inverse_tan(self, x):
    return math.atan(x)

def sinh(self, x):
    return math.sinh(x)

def cosh(self, x):
    return math.cosh(x)

def tanh(self, x):
    return math.tanh(x)

def degrees_to_radians(self, deg):
    return math.radians(deg)

def radians_to_degrees(self, rad):
    return math.degrees(rad)
```

Test Case	Expected Output	Actual Output	Remark
Calculator Class	Calculator class should work on both integer, decimals and fractional.	Calculator is working on integers , decimals and fractions.	Positive
Division function	If divided by zero then it should show a message that Cannot divide by zero.	Giving a message on dividing by zero.	Positive
Square root of negative number	Negative input should not be taken and a message should be shown	Message is showing that input must be greater than equal to 0.	Positive
Natural Logarithm and base 10 logarithm of negative numbers	Negative input should not be taken and a message should be shown that input must be greater than 0	Negative input is not working and the message is showing that input must be greater than 0.	Positive
Factorial of negative number	Negative input should not be taken	It is taking negative input and showing an error	Negative
sin(3*pi/2)	-1	-1	Positive
cos(3*pi/2)	0	1.836970198721029 7e-16	Negative
Inverse sine and cosine of a number less than -1 or greater than 1	It should not take value less than -1 and greater than 1 amd message should be shown that input must be between -1 and 1	It is taking input values less than -1 or greater than 1 and showing ValueError.	Negative

```
p = math.pi
r3 = trigno.cos(3*p/2)
print(r3)
```

[67] ✓ 0.0s

```
-1.8369701987210297e-16
```

```
p = math.pi
r3 = trigno.sin(3*p/2)
print(r3)
```

[65] ✓ 0.0s

```
-1.0
```

```
p = math.pi
r3 = trigno.inverse_sin(2)
print(r3)
```

[70] ⊗ 0.0s

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[70], line 2
      1 p = math.pi
----> 2 r3 = trigno.inverse_sin(2)
      3 print(r3)

Cell In[51], line 14
     13 def inverse_sin(self, x):
---> 14     return math.asin(x)

ValueError: math domain error
```

```
p = math.pi
r3 = trigno.inverse_sin(-2)
print(r3)
```

[72] ⊗ 0.0s

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[72], line 2
      1 p = math.pi
----> 2 r3 = trigno.inverse_sin(-2)
      3 print(r3)

Cell In[51], line 14
     13 def inverse_sin(self, x):
---> 14     return math.asin(x)

ValueError: math domain error
```

```
r2 = sc.factorial(-10)
print(r2)
```

[54] ⊗ 0.0s

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[54], line 1
----> 1 r2 = sc.factorial(-10)
      2 print(r2)

Cell In[50], line 23
     22 def factorial(self, n):
---> 23     return math.factorial(n)

ValueError: factorial() not defined for negative values
```

```
r2 = sc.natural_logarithm(-10)
print(r2)
```

[68] ✓ 0.0s

```
Input must be greater than 0 for natural logarithm
None
```

```
r2 = sc.base_10_logarithm(-10)
print(r2)
```

[69] ✓ 0.0s

```
Input must be greater than 0 for base-10 logarithm
None
```