**Assignment**

**1)User registration page :**

**reg.html :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Home</title>

<link rel="stylesheet" href="reg.css">

</head>

<body>

<div class="sub-body">

<header>

<h1><center>Registration Form</center></h1>

</header>

<main>

<div class="reg">

<form class="cls">

<div class="input-icons">

<center>

<b><label for="name"> UserName:</label></b>

<i class="fa fa-user icon"></i>

<input type="text" class="input" placeholder="UserName" id="user" autocomplete="off" required><br><br>

<b><label for="password">Password:</label></b>

<i class="fa fa-lock icon"></i>

<input type="password" class="input" placeholder="Password" id="pass"required><br><br>

<b><label for="email"> Email Id:</label></b>

<i class="fa fa-envelope icon"></i>

<input type="email" class="input" placeholder="Email" id="mail"required><br><br>

<button type="button" id="myBtn" class="but">Submit</button>

</center>

</div>

</div>

</form>

</main>

</div>

</body>

</html>

**reg.css**

.container{

    padding: 250px;

    background-color: lightblue;

}

input{

    width: 100%;

    padding: 12px;

    margin: 3px 0 12px 0;

    display: inline-block;

    border: none;

    background: #f1f1f1;

}

input:focus{

    background-color: rgb(0, 179, 255);

    outline: none;

}

hr {

    border: 1px solid #f1f1f1;

    margin-bottom: 10px;

}

.registerbtn{

    background-color: #4CAF50;

    color: white;

    padding: 10px 20px;

    margin: 8px 0;

    border: none;

    cursor: pointer;

    width: 100%;

    opacity: 0.9;

}

.registerbtn:hover{

    opacity: 1;

}

**Output :**



**2)HTML Formatting Elements**

Formatting elements were designed to display special types of text:

* <b> - Bold text

<html>

<body>

<p><b>This text is bold.</b></p>

</body>

</html>

* <strong> - Important text

<p><strong>This text is important!</strong></p>

<i> - Italic text

<p><i>This text is italic.</i></p>

* <em> - Emphasized text

<p><em>This text is emphasized.</em></p>

* <mark> - Marked text

<p>Do not forget to buy <mark>pencil</mark> today.</p>

* <small> - Smaller text

<p><small>This is some smaller text.</small></p>

* <del> - Deleted text

<p>My favorite color is <del>black</del> red.</p>

* <ins> - Inserted text

<p>My favorite color is <del>black</del> <ins>purple</ins>.</p>

* <sub> - Subscript text

<p>This is <sub>subscripted</sub> text.</p>

* <sup> - Superscript text

<p>This is <sup>superscripted</sup> text.</p>

**3)HTML Input Types**

Here are the different input types you can use in HTML:

* <input type="button">
* <input type="checkbox">
* <input type="color">
* <input type="date">
* <input type="datetime-local">
* <input type="email">
* <input type="file">
* <input type="hidden">
* <input type="image">
* <input type="month">
* <input type="number">
* <input type="password">
* <input type="radio">
* <input type="range">
* <input type="reset">
* <input type="search">
* <input type="submit">
* <input type="tel">
* <input type="text">
* <input type="time">
* <input type="url">
* <input type="week">

**Module 3:**

* Consider a list (list=[]). You can perform the following commands: insert e: Insert integer at position. print: Print the list. remove e: Delete the first occurrence of integer. append e: Insert integer at the end of the list. sort: Sort the list. pop: Pop the last element from the list. reverse: Reverse the list.

Initialize your list and read in the value of followed by lines of commands where each command will be of the types listed above from the list. reverse: Reverse the list. Initialize your list and read in the value of followed by lines of commands where each command will be of the types listed above. Iterate through each command in order and perform the corresponding operation on your list.

**Sample Input:**

12

insert 0 5

insert 1 10

insert 0 6

print

remove 6

append 9

append 1

sort

print

pop

reverse

print

**Program :**

import sys

if \_name\_ == '\_main\_':

N = int(input())

my\_list = []

inputs = []

for line in sys.stdin:

inputs.append(line)

for item in inputs:

if item[0:5] == 'print':

print(my\_list)

elif item[0:2] == 'in':

inserts = [s for s in item.split()][1:3]

inserts = list(map(int, inserts))

my\_list.insert(inserts[0], inserts[1])

elif item[0:3] == 'rem':

inserts = list(map(int, [s for s in item.split()][1]))

my\_list.remove(inserts[0])

elif item[0:2] == 'ap':

inserts = list(map(int, [s for s in item.split()][1]))

my\_list.append(inserts[0])

elif item[0:4] == 'sort':

my\_list.sort()

elif item[0:3] == 'pop':

my\_list.pop()

elif item[0:7] == 'reverse':

my\_list.reverse()

**output :**

[6, 5, 10]

[1, 5, 9, 10]

[9, 5, 1]

**2)Simple calculator**

**Program:**

def add(x, y):

return x + y

def subtract(x, y):

return x - y

def multiply(x, y):

return x \* y

def divide(x, y):

return x / y

print("Select operation.")

print("1.Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

while True:

choice = input("Enter choice(1/2/3/4): ")

if choice in ('1', '2', '3', '4'):

num1 = float(input("Enter first number: "))

num2 = float(input("Enter second number: "))

if choice == '1':

print(num1, "+", num2, "=", add(num1, num2))

elif choice == '2':

print(num1, "-", num2, "=", subtract(num1, num2))

elif choice == '3':

print(num1, "\*", num2, "=", multiply(num1, num2))

elif choice == '4':

print(num1, "/", num2, "=", divide(num1, num2))

next\_calculation = input("Let's do next calculation? (yes/no): ")

if next\_calculation == "no":

break

else:

print("Invalid Input")

3)Write a program for concatenate,reverse ,slice a string?

**Concatenate:**

Str1=”Haiii”

Str2=”world”

Str3=str1+str2

Print(str3)

**Reverse:**

txt = "Haiii World"[::-1]

print(txt)

**slice :**

b = "Hello, World!"

print(b[:5])

**output:**

Haiii

4)why is python a popular programming language?

Due to its ease of learning and usage, Python codes can easily be written and executed much faster than other programming languages. One of the main reasons why Python’s popularity has exponentially grown is due to its simplicity in syntax so that it could be easy to read and developed by amateur professionals as well

Python is Handy for Web Development Purposes has Multiple Libraries and Frameworks

5)what are the other frameworks in python?

* Django
* Pyramid
* Turbogears
* Web2py
* Cherrypie
* Flask
* Sanic

6)Full form of WSGI

The Web Server Gateway Interface is standard interface between web server software and web applications written in Python. Having a standard interface makes it easy to use an application that supports WSGI with a number of different web servers.