**Practical No 10**

**Code:-**

<html>

<head>

<title>Registration Form </title>

</head>

<body bgcolor="aqua"><center>

<h1>ENTRY FORM</h1></center>

<form name=form1 >

<table name=tab cellspacing=15px>

<tr><td align=left><h2>Enter your Name :</h2> </td><td align=right><input

type=text name=t1 size=50>

<tr><td align=left><h2>Enter your Age :</h2> </td><td align=right><input

type=text name=t2 maxlength=3 size=50>

<tr><td align=left><h2>Enter your Address :</h2> </td><td align=right><textarea

name=ta rows=6 cols=45></textarea>

<tr><td align=left><h2>Gender:</h2> </td><td align=left><input type=radio name=r1

value="female">Female<br>

<input type=radio name=r1 value=male>Male</td>

<tr><td align=left><h2>Languages Known :</h2> </td><td

align=left><center>(select more than one)</center>

<input type=checkbox name=c1 value=c>C<br>

<input type=checkbox name=c2 value=c++>C++<br>

<input type=checkbox name=c3 value=vb>VB<br>

<input type=checkbox name=c4 value=java>JAVA<br>

<input type=checkbox name=c5 value=asp>ASP<br>

<input type=checkbox name=c6 value=others>OTHERS<br></td>

<tr><td align=left><h2>Enter your Password :</h2> </td><td align=right><input

type=password name=t3 size=50>

</table><center><input type=reset value=" Reset " >

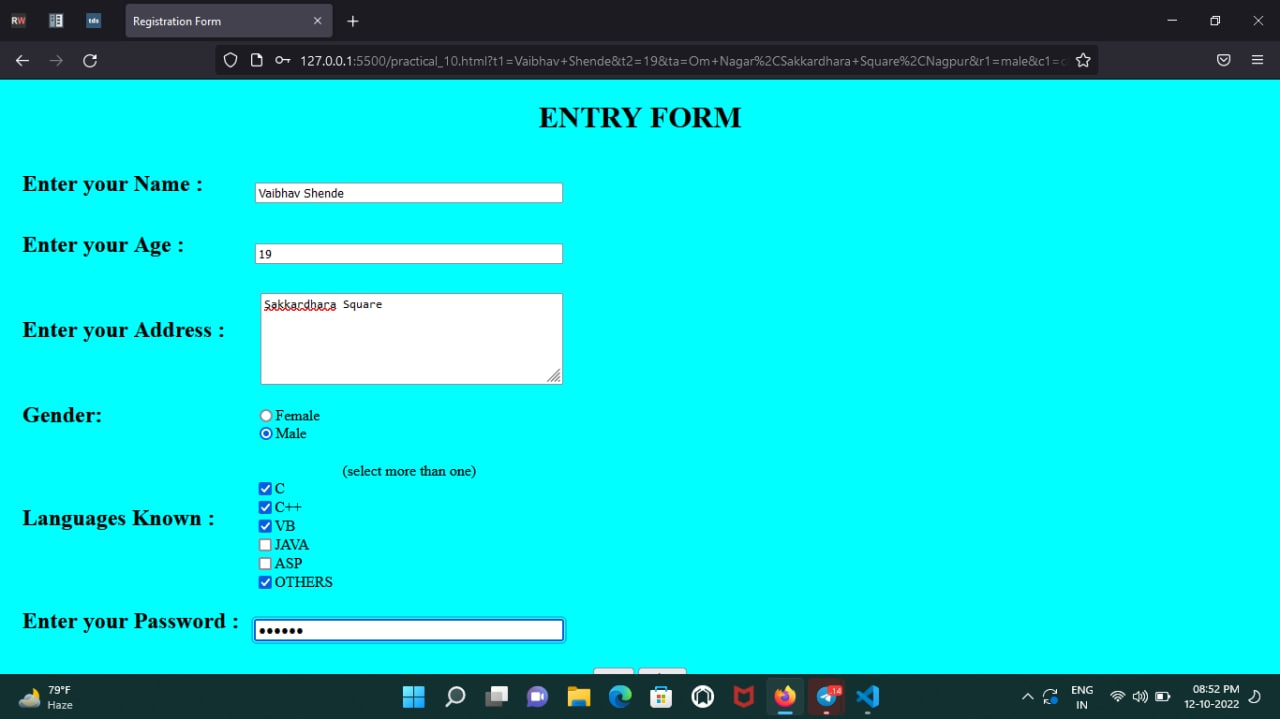
<input type=submit value=" Submit " >

</form>

</body>

</html>

**Output:-**



**PRACTICAL NO.11**

**CODE:-**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Practical10</title>

<script>

function check() {

var detailBox = document.myForm.detailBox.value;

var countOfWords = detailBox.split(" ");

var k = detailBox.split('\n');

alert("total no of characters are : " + detailBox.length +

"\ntotal no of words are :" + countOfWords.length + "\ntotal no of new lines are : " + k.length);

}

</script>

</head>

<body>

<form name="myForm" onsubmit="check()">

<textarea id="detailBox" cols="50" rows="10"></textarea><br>

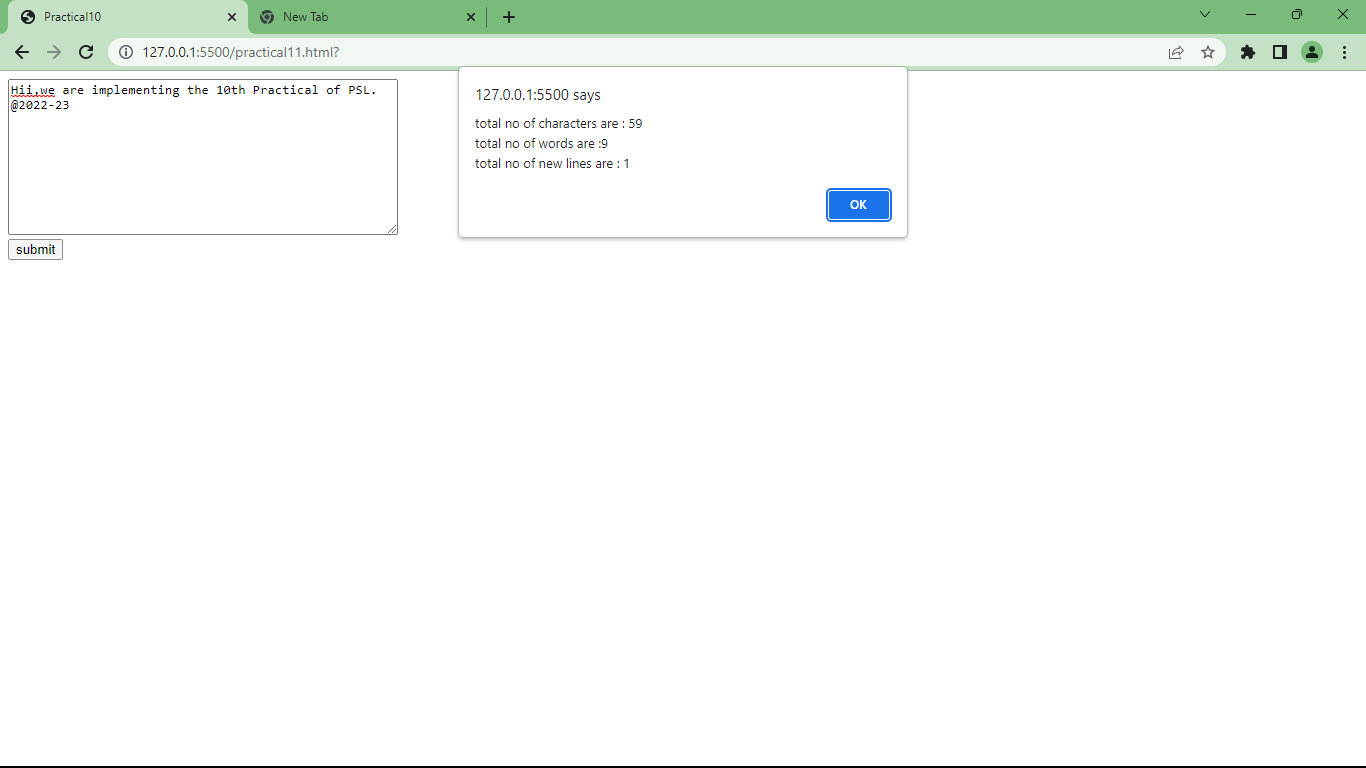
<input type="submit" value="submit">

</form>

</body>

</html>

**OUTPUT:-**



**Practical No.08**

**Code:-**

**Main.html**

<html>

<head>

<title>Web Appln</title>

</head>

<body>

<center>

<form method="post" action="webApp.html">

<h2 align="center">Web Application</h2>

<br><br>

<label>Enter Your Name :</label>

<input type="text" name="uname">

<br> <br><br>

<label>Enter Your Age :</label>

<input type="text" name="age">

<br><br><br>

<input type="submit" value="Submit"></input>

</center>

</form>

</body>

</html>

**webapp.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Document</title>

</head>

<body>

<?php

$name=$\_POST["uname"];

$age=$\_POST["age"];

if($age<18)

{echo "Hello ".$name." , <font color=red size=5>You are not authorized to visit this site</font>";}

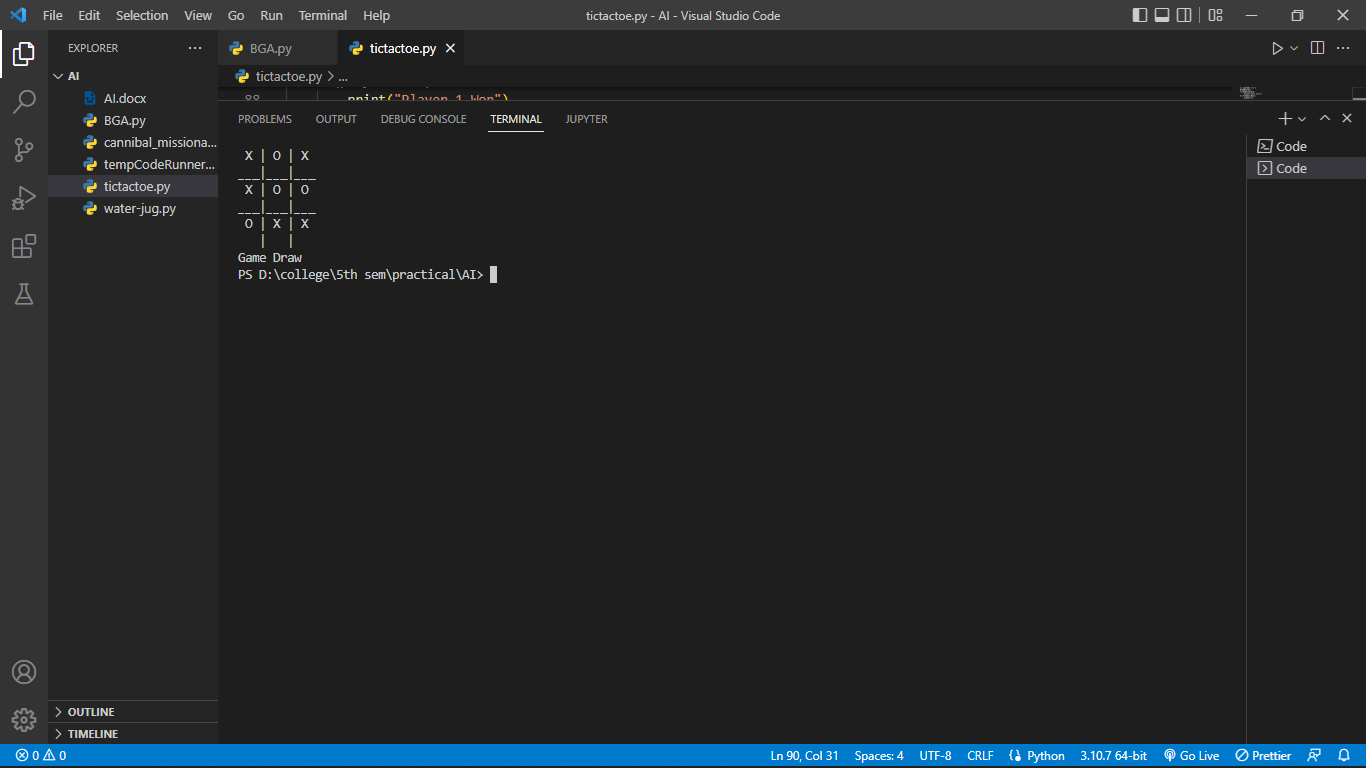
else{ echo "Hello ".$name." ,<font color=green size=5> Welcome to this site</font>";}

?>

</body>

</html>

**Output:-**



**Practical No. 11**

**Code:**

import random

POPULATION\_SIZE = 100

GENES = '''abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOP

QRSTUVWXYZ 1234567890, .-;:\_!"#%&/()=?@${[]}'''

TARGET = "I love Artificial Intelligence"

class Individual(object):

'''

Class representing individual in population

'''

def \_\_init\_\_(self, chromosome):

self.chromosome = chromosome

self.fitness = self.cal\_fitness()

@classmethod

def mutated\_genes(self):

'''

create random genes for mutation

'''

global GENES

gene = random.choice(GENES)

return gene

@classmethod

def create\_gnome(self):

'''

create chromosome or string of genes

'''

global TARGET

gnome\_len = len(TARGET)

return [self.mutated\_genes() for \_ in range(gnome\_len)]

def mate(self, par2):

'''

Perform mating and produce new offspring

'''

child\_chromosome = []

for gp1, gp2 in zip(self.chromosome, par2.chromosome):

prob = random.random()

if prob < 0.45:

child\_chromosome.append(gp1)

elif prob < 0.90:

child\_chromosome.append(gp2)

else:

child\_chromosome.append(self.mutated\_genes())

return Individual(child\_chromosome)

def cal\_fitness(self):

'''

Calculate fitness score, it is the number of

characters in string which differ from target

string.

'''

global TARGET

fitness = 0

for gs, gt in zip(self.chromosome, TARGET):

if gs != gt:

fitness += 1

return fitness

def main():

global POPULATION\_SIZE

generation = 1

found = False

population = []

for \_ in range(POPULATION\_SIZE):

gnome = Individual.create\_gnome()

population.append(Individual(gnome))

while not found:

population = sorted(population, key=lambda x: x.fitness)

if population[0].fitness <= 0:

found = True

break

new\_generation = []

s = int((10\*POPULATION\_SIZE)/100)

new\_generation.extend(population[:s])

s = int((90\*POPULATION\_SIZE)/100)

for \_ in range(s):

parent1 = random.choice(population[:50])

parent2 = random.choice(population[:50])

child = parent1.mate(parent2)

new\_generation.append(child)

population = new\_generation

print("Generation: {}\tString: {}\tFitness: {}".

format(generation,

"".join(population[0].chromosome),

population[0].fitness))

generation += 1

print("Generation: {}\tString: {}\tFitness: {}".

format(generation,

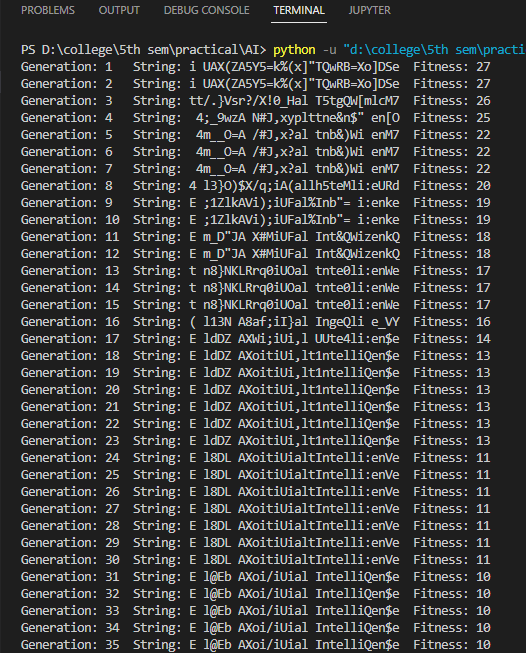
"".join(population[0].chromosome),

population[0].fitness))

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

main()

**Output:**

****