MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT7412 – WEB TECHNOLOGY LABORATORY

LAB MANUAL

REGULATION - 2015

Vision of the Department

To educate students with conceptual knowledge and technical skills in the field of Information Technology with moral and ethical values to achieve excellence in an academic, industry and research centric environment.

Mission of the Department

1. To inculcate in students a firm foundation in theory and practice of IT skills coupled with the thought process for disruptive innovation and research methodologies, to keep pace with emerging technologies.
2. To provide a conducive environment for all academic, administrative, and interdisciplinary research activities using state-of-the-art technologies.
3. To stimulate the growth of graduates and doctorates, who will enter the workforce as productive IT engineers, researchers, and entrepreneurs with necessary soft skills, and continue higher professional education with competence in the global market.
4. To enable seamless collaboration with the IT industry and Government for consultancy and sponsored research.
5. To cater to cross-cultural, multinational, and demographic diversity of students.
6. To educate the students on the social, ethical, and moral values needed to make significant contributions to society.

Program Educational Objectives (PEO)

After completion of the B.Tech. (IT) course, students will be able to:

PEO1: Demonstrate core competence in basic engineering and mathematics to design, formulate, analyse, and solve hardware/ software engineering problems.

PEO2: Have insight in fundamental areas of Information Technology and related engineering with an inclination towards self-learning to address real-world problems using digital and cognitive technologies.

PEO3: Collaborate with industry, academic and research institutions for product and research related development.

PEO4: Imbibe high professionalism, effective communication skills and team spirit to work on multidisciplinary projects, in diverse professional environments.

PEO5: Practice IT solutions following technical standards with ethical values.

Program Specific Outcomes(PSO)

PSO1: To apply programming principles and practices for the design of software solutions in an internet-enabled world of business and social activities.

PSO2: To identify the resources to build and manage the IT infrastructure using the current technologies in order to solve real world problems with an understanding of the tradeoffs involved in the design choices.

PSO3: To plan, design and execute projects for the development of intelligent systems with a focus on the future

IT7412 **WEB TECHNOLOGY LABORATORY** **L T P C**

0 0 4 2

OBJECTIVES:

- To learn about web technologies related concepts
- To develop Java and HTML based web applications
- To implement parsers and XML related concepts.

Exercises

1. Creating simple applications using JAVA by exploring all the object oriented programming
concepts such as inheritance, polymorphism, interfaces and packages.
2. Creating GUI based application using JAVA Swings
3. Developing concurrent and generic programming using Threads
4. Creation of simple websites using HTML 5 Tags
5. Creation of web forms and validating it through javascripts
6. Creation of XML file and validating with DTD and XML schema
7. Working with DOM and SAX parsers
8. Creation of AJAX based application
9. Developing JSON application
10. Creation of dynamic HTML based web applications
11. Creation of servlet based web application with JDBC
12. Developing JSP application
13. Creating simple applications using python
14. Simple database and web application using python

TOTAL: 60 PERIODS

OUTCOMES:

On Completion of the course, the students should be able to:

- Create simple web applications
- Implement server side and client side programming develop web applications with various web technology concepts.

Mapping of Course Outcomes (COs) with Program Outcomes (POs)

CO	Program Outcomes(POs)												PSO1	PSO2	PSO3
	1	2	3	4	5	6	7	8	9	10	11	12			
Create simple web applications	1	2	3	1	1	1	0	0	0	0	1	1	2	2	2
Write programs on streaming, multithreading and generic	1	2	3	2	2	0	0	0	0	0	0	1	2	2	2

collections in java															
Work with xml based technologies	**1**	**2**	**2**	**3**	**3**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**2**	**2**	**2**
Develop web based applications using jdbc	**1**	**2**	**2**	**3**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**1**	**2**	**2**	**2**
Implement server side programming using servlet and jsp	**1**	**2**	**2**	**3**	**3**	**2**	**2**	**0**	**2**	**1**	**3**	**1**	**2**	**2**	**2**
Develop simple web application using python	**1**	**2**	**2**	**3**	**3**	**3**	**3**	**1**	**2**	**1**	**3**	**1**	**2**	**2**	**2**

GRADING RUBRIC FOR LABORATORY COURSES			
	Good Marks (81%-100%)	**Average Marks (50%-80%)**	**Satisfactory Marks (< 50%)**
Continuous Assessment (Covers Preparedness, Basic implementation, Ability to adapt additional features and coding standards) **(Max Marks:25)**	Presence of detailed procedure, coding samples with proper implementation. Able to adapt the changes in the code quickly. Proper Coding Style.	Clarity of the procedure and coding samples are average with partial implementation. Able to understand the changes but unable to implement it. Fairly presented code with medium standards.	Lack of detailed procedure as well as coding samples with incorrect implementation. Unable to adapt the changes in coding. Coding standards are not followed. Code is messy.
Laboratory Test (Covers Understanding of problem, Basic Problem Solving and Ability to code, test, run and debug within the stipulated time) **(Max Marks:25)**	Problem understood clearly and solved. Complete implementation with proper test data within the stipulated time.	Problem understood but problem solving is not full-fledged. Completion of three fourths of the implementation with proper test data.	Lack of understanding and problem-solving ability is poor. Implementation not completed/ Partial implementation within the stipulated time.
Course Oriented Laboratory Project (Covers Problem Selection, Demonstration of the Project, Wide coverage of concepts in the target language) **(Max Marks:25)**	Selection of good real time problem with Complete implementation with in-depth understanding on the concepts implemented. Wide coverage of concepts in the target language.	Selection of good real time problem with partially complete implementation and good knowledge on the concepts implemented. Moderate coverage of concepts.	Selection of fair problem with incomplete implementation. Lack of proper knowledge and understanding on the concepts implemented. Limited coverage of concepts.

S.NO	NAME OF THE EXPERIMENT
1.	Java Introduction
2.	Java IO Statements
3.	Java Constructors
4.	Java Inheritance
5.	java Polymorphism and interfaces
6.	Java Arrays
7.	Java Collections
8.	Java Swings GUI
9.	Java Threading
10.	Simple Websites using HTML5 Tags
11.	Web Form Creation and Validation using Java Scripts
12.	Simple Servlet in Java
13.	Servlet Web based Application with JDBC
14.	Developing JSP Application
15	Simple Applications using Python
16	Database with Python – SQLite3

EX.NO: 1 **JAVA INTRODUCTION**

AIM

To work with basic java programs.

1.FACTORIAL OF A NUMBER

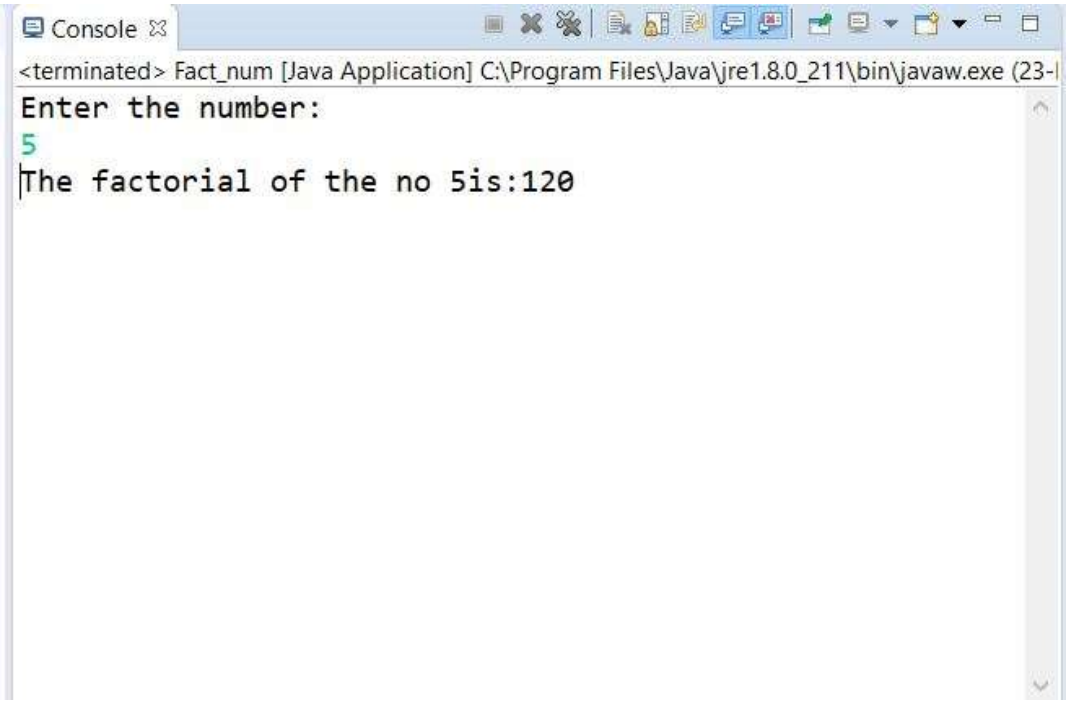
ALGORITHM

1: Read number n.
2. Initialize i and fact to 1.
3. Repeat step 4 and step 5 while i is not equal to n.
4. fact <- fact * i
5. i <- i +1
6. Return fact

SOURCE CODE:

```
package class_and_objects;
import java.util.*;
public class Fact_num {

        public static void main(String[] args) {
                 int n,f=1,c;
                 System.out.println("Enter the number:");
                Scanner in=new Scanner(System.in);
                n=in.nextInt();
                if(n<0)
                     System.out.println("The no is negative");
                else
                   for(c=1;c<=n;c++)
                      f=f*c;
                 System.out.println("The factorial of theno"+n+"is:"+f);

        }

}
```

OUTPUT



II PRIME NUMBER

ALGORITHM

1: Start

2: Read number n

3: Set f=0

4: For i=2 to n-1

5: If n mod 1=0 then

6: Set f=1 and break

7: Loop

8: If f=0 then

print 'The given number is prime'

else

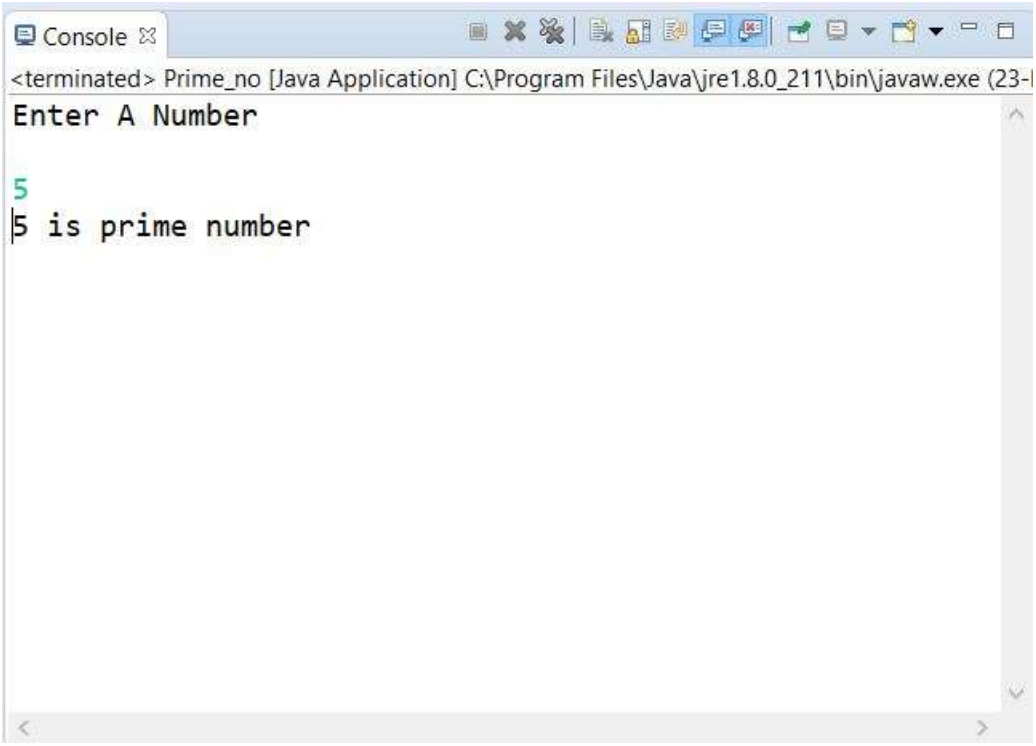
print 'The given number is not prime'

9: Stop

SOURCE CODE

```
package class_and_objects;
import java.util.*;
public class Prime_no {

        public static void main(String[] args) {
                int n,m=0,flag=0,i;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter A Number");
        n =s.nextInt();
        m=n/2;
        if(n==0||n==1){
         System.out.println(n+" is not prime number");
        }else{
         for(i=2;i<=m;i++){
          if(n%i==0){
           System.out.println(n+" is not prime number");
           flag=1;
           break;
          }
         }
         if(flag==0)  { System.out.println(n+" is prime number"); }
        }
         }

}
```

OUTPUT



III REVERSE OF A NUMBER

ALGORITHM

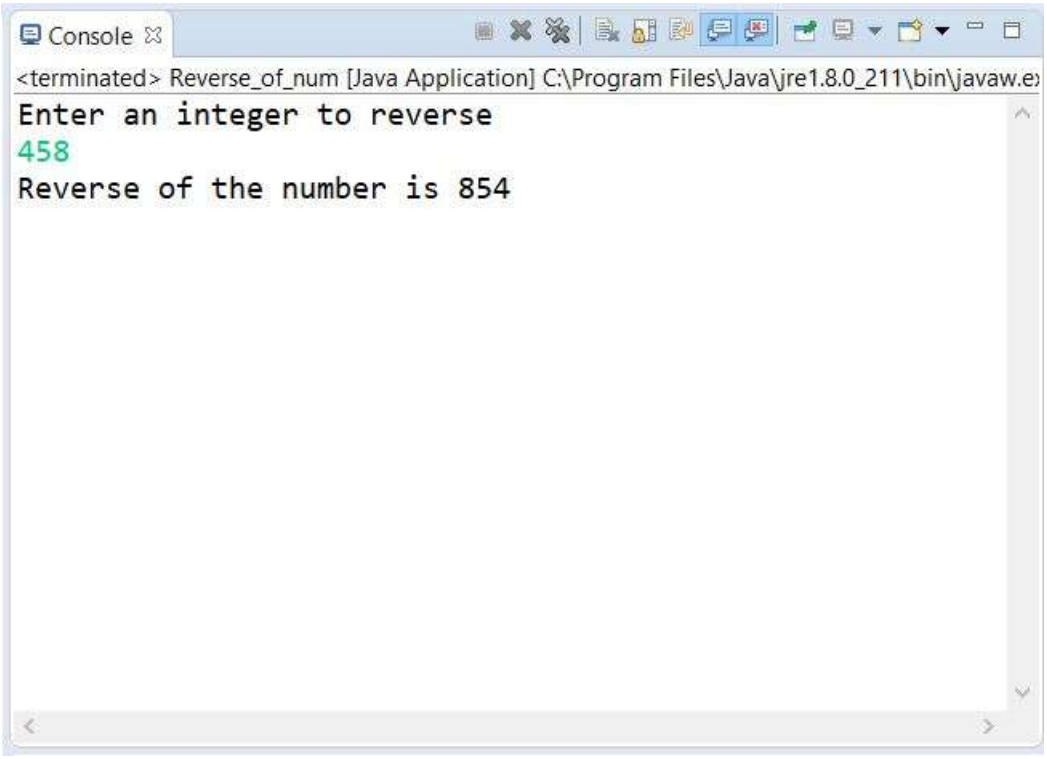
Input: num
1: Initialize rev_num = 0
2: Loop while num > 0
Multiply rev_num by 10 and add remainder of num
divide by 10 to rev_num
rev_num = rev_num*10 + num%10;
Divide num by 10
3: Return rev_num

SOURCE CODE

```
package class_and_objects;
import java.util.*;
public class Reverse_of_num {

        public static void main(String[] args) {
                int n, r = 0;

                    System.out.println("Enter an integer to reverse");
                    Scanner in = new Scanner(System.in);
                    n = in.nextInt();

                    while(n != 0)
```

```
                {
                  r = r * 10;
                  r = r + n%10;
                  n = n/10;
                }

                System.out.println("Reverse of the number is " + r);

        }

}
```

OUTPUT



IV ARMSTRONG NUMBER

ALGORITHM

1:read number

2:set sum=0 and duplicate=number

3:while number > 0

4:reminder=number%10

5:sum=sum+(reminder*reminder*reminder)

6:number=number/10

7:if sum = duplicate

8:display number is armstrong

9:else

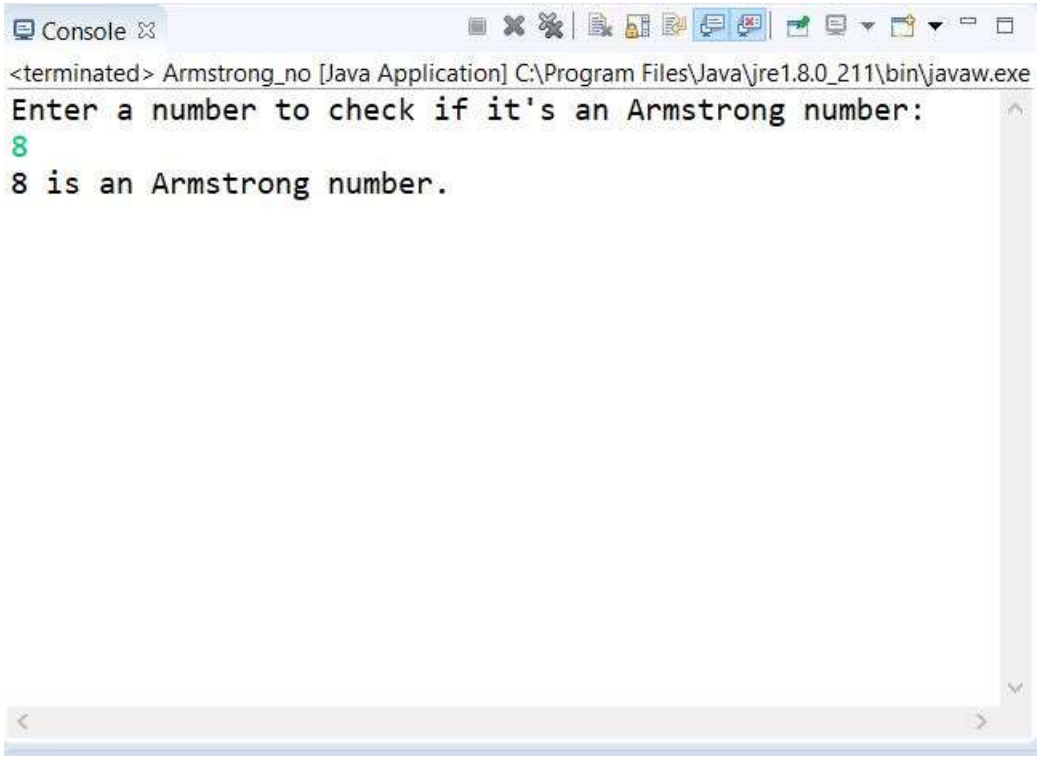
10:display number is not armstrong

SOURCE CODE

```
 package class_and_objects;
import java.util.*;
public class Armstrong_no {

        public static void main(String[] args) {
                int n, sum = 0, count, remainder, digits = 0;

            Scanner in = new Scanner(System.in);
            System.out.println("Enter a number to check if it's an Armstrong
number:");
            n = in.nextInt();

            count = n;


            while (count != 0) {
              digits++;
              count = count/10;
            }
           count = n;

            while (count != 0) {
              remainder = count%10;
              sum = sum + power(remainder, digits);
              count = count/10;
            }

            if (n == sum)
              System.out.println(n + " is an Armstrong number.");
            else
              System.out.println(n + " isn't an Armstrong number.");
          }

          static int power(int n, int r) {
            int c, p = 1;
```

```
            for (c = 1; c <= r; c++)
              p = p*n;

            return p;

        }

}
```

OUTPUT



V SUM OF DIGITS

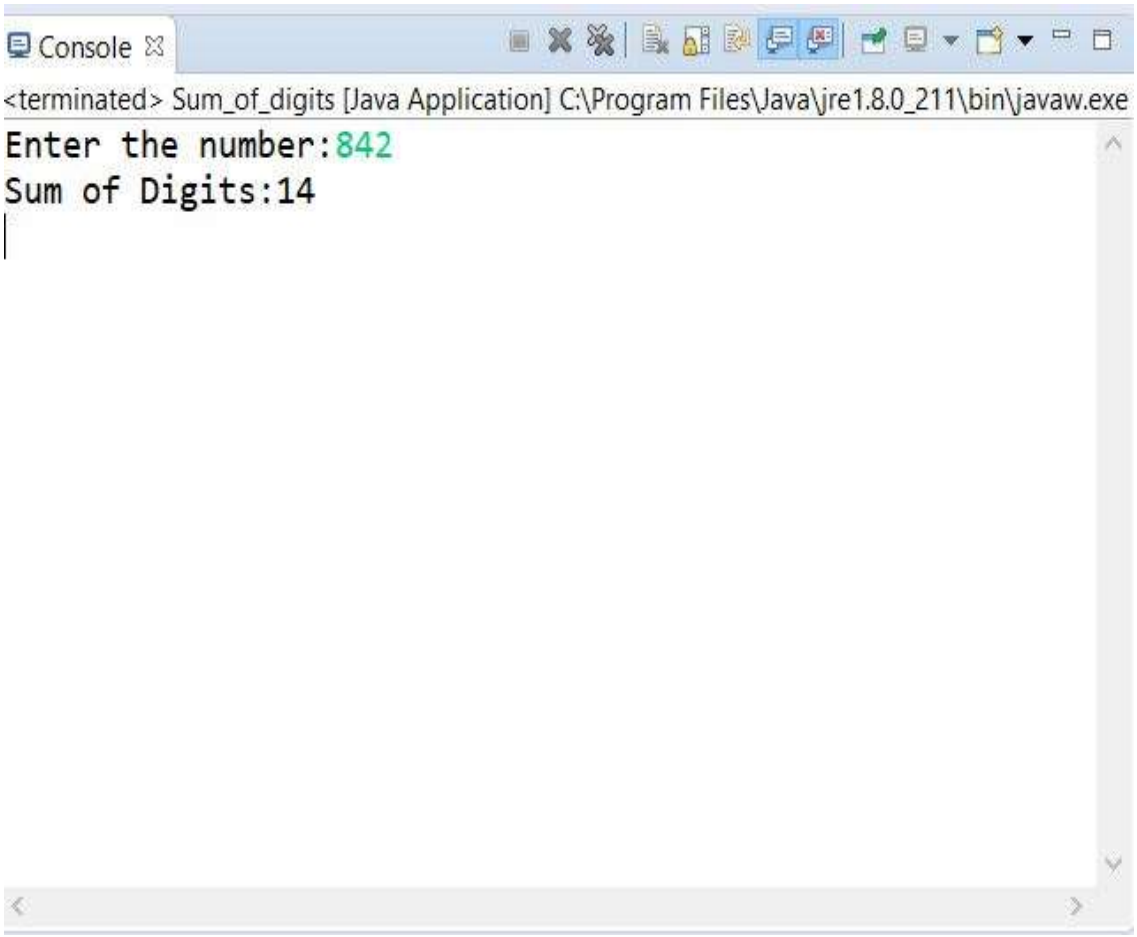
ALGORITHM

1. Input a Number
2. Initialize Sum to zero
3. While Number is not zero
   Get Remainder by Number Mod 10
   Add Remainder to Sum
   Divide Number by 10
4. Print sum

SOURCE CODE

```
package class_and_objects;
import java.util.*;
public class Sum_of_digits {

	public static void main(String[] args) {
		  int m, n, sum = 0;
		Scanner s = new Scanner(System.in);
		System.out.print("Enter the number:");
		m = s.nextInt();
		while(m > 0)
		{
		   n = m % 10;
		   sum = sum + n;
		   m = m / 10;
		}
		System.out.println("Sum of Digits:"+sum);

	}

}
```

OUTPUT



VI PALINDROME OF A NUMBER

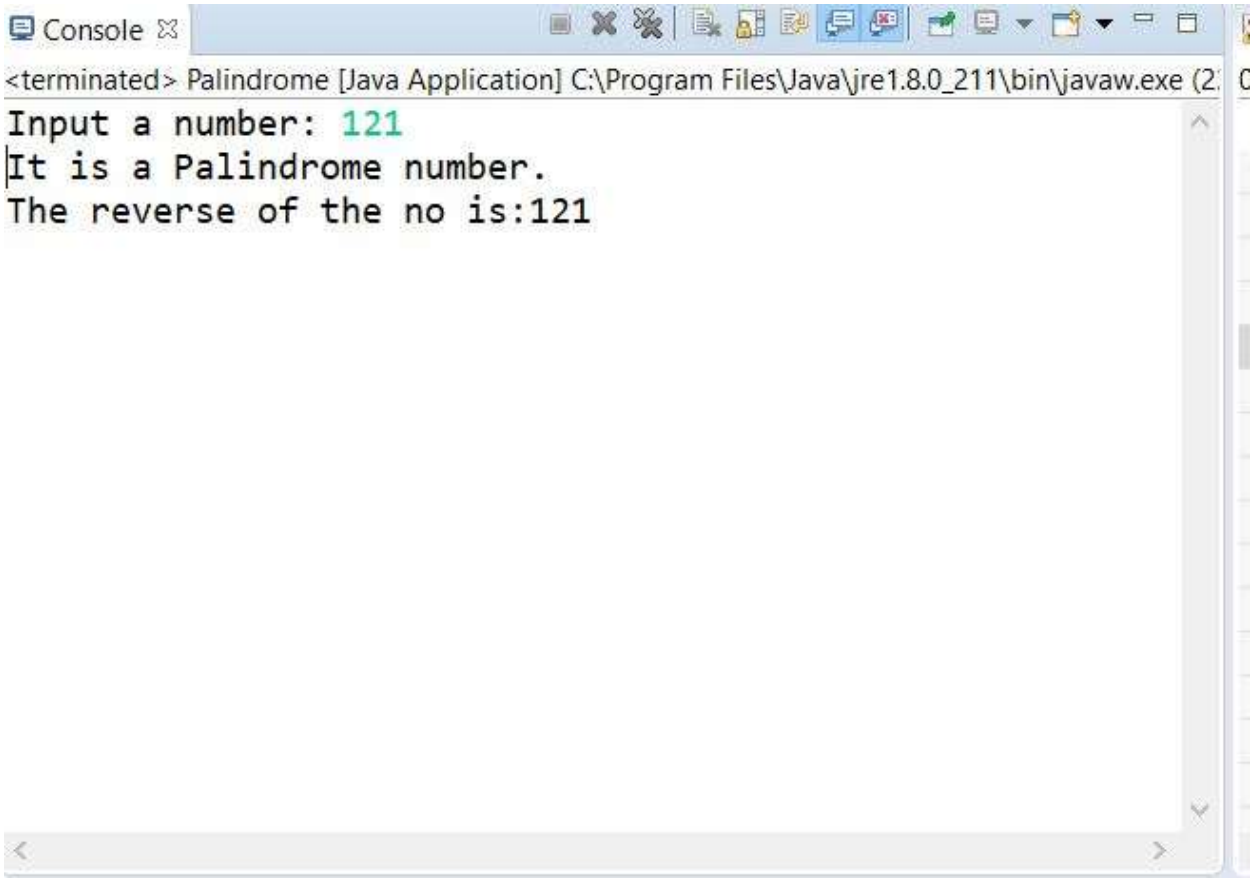
ALGORITHM

1:Start
2:get a number(x)
3:set p=0 and temp=x
4:divide x by 10, store reminder (x%10) in r
5:set x=x/10
6:set p= p + r
7:set p=p*10
8:repeat from step 4 till x is equal to 0
9: set p = p /10
10: check if p is equal to temp
11: if so, print palindrome
12: else print not a palindrome
13:Stop

SOURCE CODE

```
package class_and_objects;
import java.util.*;
public class Palindrome {

        public static void main(String[] args) {
                Scanner in = new Scanner(System.in);
            System.out.print("Input a number: ");
            int n = in.nextInt();
            int sum = 0, r;
                 int temp = n;
            while(n>0)
                   {
              r = n % 10;
              sum = (sum*10)+r;
              n = n/10;
             }
             if(temp==sum)
              System.out.println("It is a Palindrome number.");
             else
              System.out.println("Not a palindrome");

             System.out.println("The reverse of the no is:"+sum);
        }

}
```

OUTPUT



RESULT

Thus examples of basic java programs are successfully implemented.

EX.NO: 2 **JAVA IO STATEMENTS**

AIM

To create java programs using various methods of input arguments.

I. STATIC ASSIGNMENT INPUT

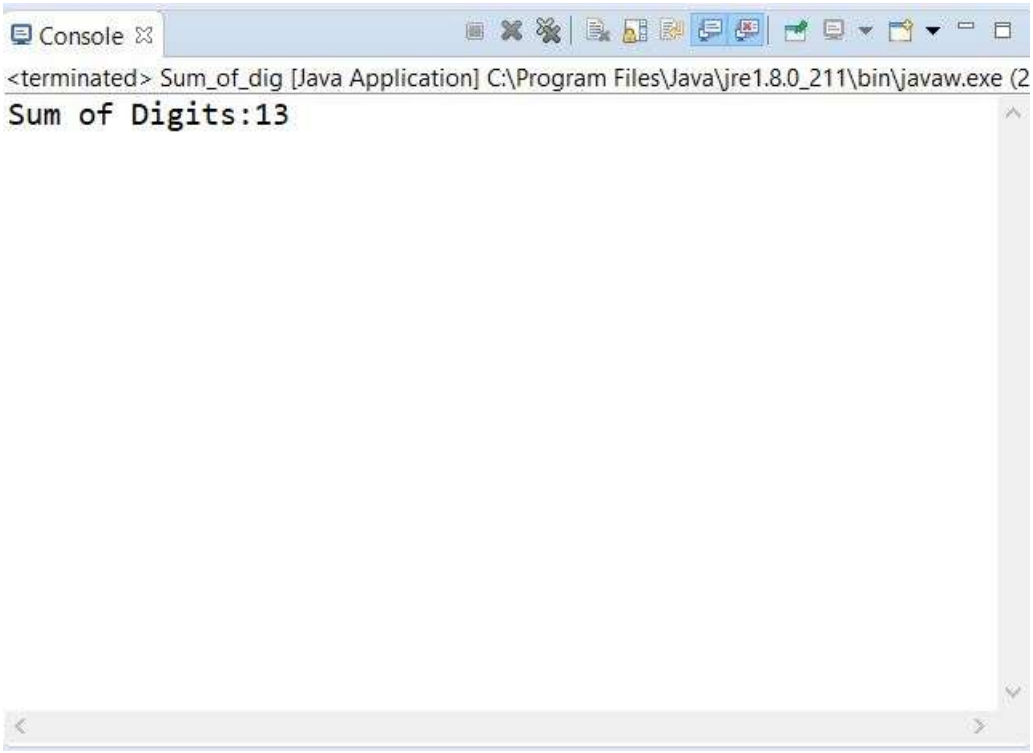
ALGORITHM

1:Initialize a number to the input needed
2: Initialize Sum to zero
3:While Number is not zero
Get Remainder by Number Mod 10
Add Remainder to Sum
Divide Number by 10
4:Print sum

SOURCE CODE

```
package class_and_objects;


public class Sum_of_dig {

        public static void main(String[] args) {
                int m=562, n, sum = 0;

       while(m > 0)
       {
          n = m % 10;
          sum = sum + n;
          m = m / 10;
       }
       System.out.println("Sum of Digits:"+sum);

        }

}
```

OUTPUT



II.DYNAMIC INPUT EXAMPLE

ALGORITHM

1:read number

2:set sum=0 and duplicate=number

3:while number > 0

4:reminder=number%10

5:sum=sum+(reminder*reminder*reminder)

6:number=number/10

7:if sum = duplicate

8:display number is armstrong

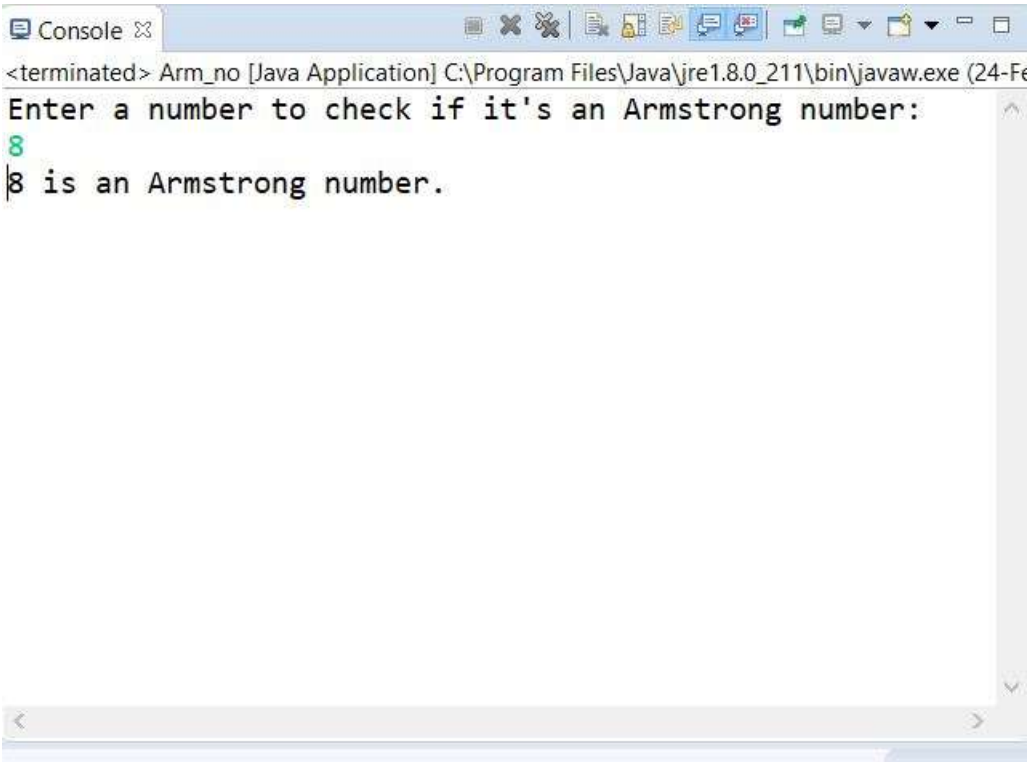
9:else

10:display number is not armstrong

SOURCE CODE

```
package class_and_objects;

import java.io.*;

public class Arm_no {


        public static void main(String[] args) throws IOException {
                int n, sum = 0, count, remainder, digits = 0;

            DataInputStream in = new DataInputStream(System.in);
            System.out.println("Enter a number to check if it's an Armstrong
number:");
            n = Integer.parseInt( in.readLine());

            count = n;


            while (count != 0) {
              digits++;
              count = count/10;
            }
           count = n;

            while (count != 0) {
              remainder = count%10;
              sum = sum + power(remainder, digits);
              count = count/10;
            }

            if (n == sum)
              System.out.println(n + " is an Armstrong number.");
            else
              System.out.println(n + " isn't an Armstrong number.");
          }

          static int power(int n, int r) {
            int c, p = 1;

            for (c = 1; c <= r; c++)
              p = p*n;

            return p;}}
```

OUTPUT



III. COMMAND LINE ARGUMENT EXAMPLE

ALGORITHM

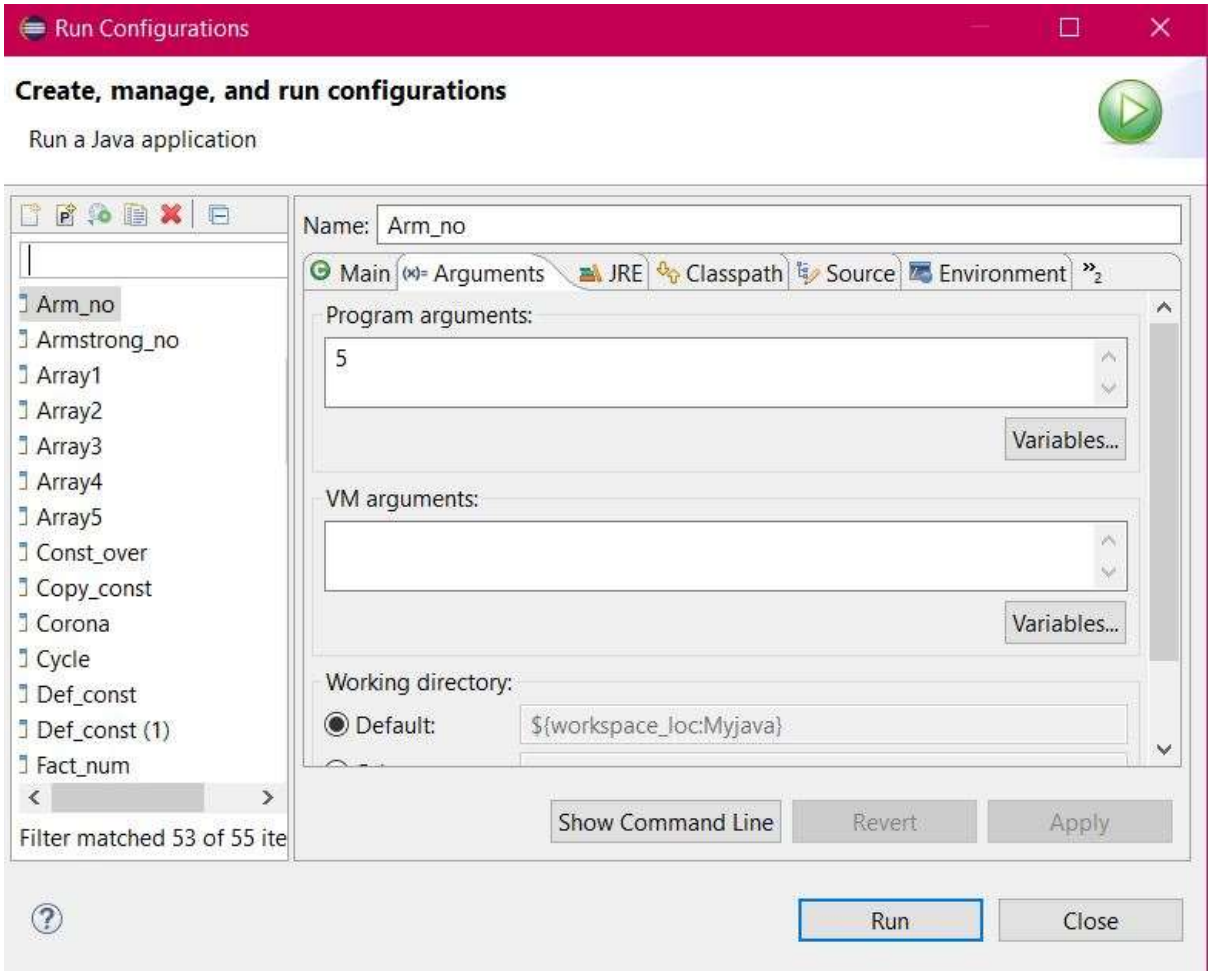
1: Input a number using command line argument
2. Initialize i and fact to 1.
3. Repeat step 4 and step 5 while i is not equal to n.
4. fact <- fact * i
5. i <- i +1
6. Return fact

SOURCE CODE

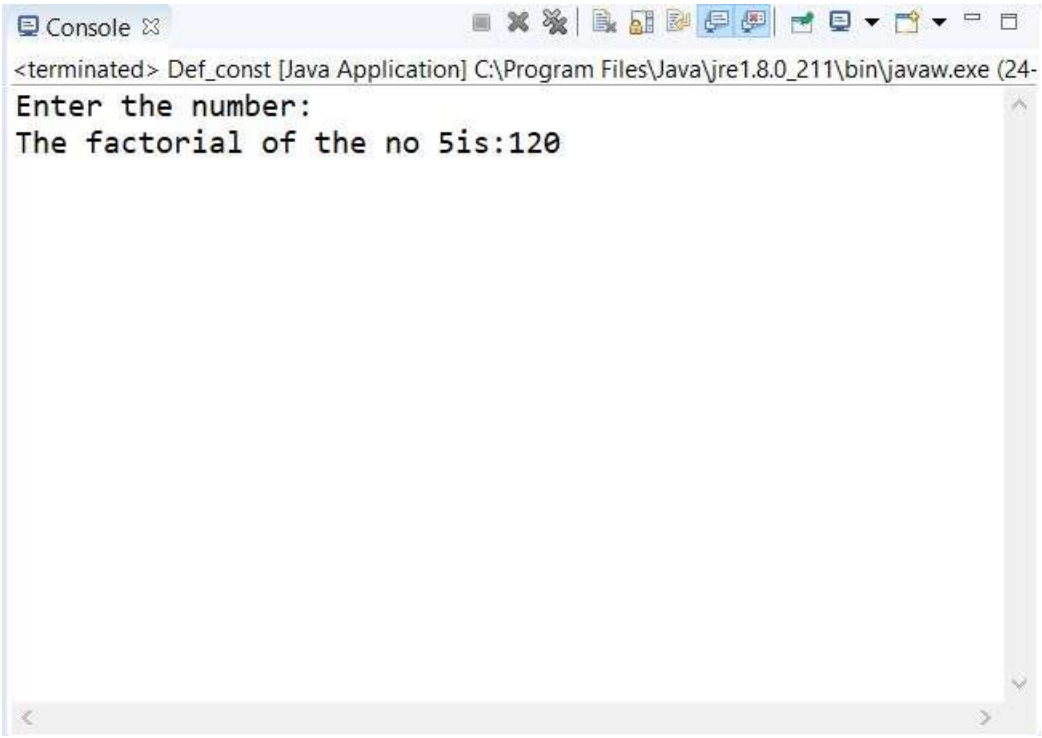
```
package class_and_objects;


public class Fact_args {

        public static void main(String[] args) {
                int n,f=1,c;
                System.out.println("Enter the number:");
                 n=Integer.parseInt(args[0]);
               if(n<0)
                   System.out.println("The no is negative");
```

```
            else
               for(c=1;c<=n;c++)
                  f=f*c;
             System.out.println("The factorial of the no "+n+"is:"+f);


        }
}
```

OUTPUT



OUTPUT – (Continue)



RESULT

Thus java IO programs are implemented and executed successfully.

EX.NO: 3 **JAVA CONSTRUCTORS**

AIM

To implement the types of constructors using java programs.

I. DEFAULT CONSTRUCTORS

ALGORITHM

1.Start the program

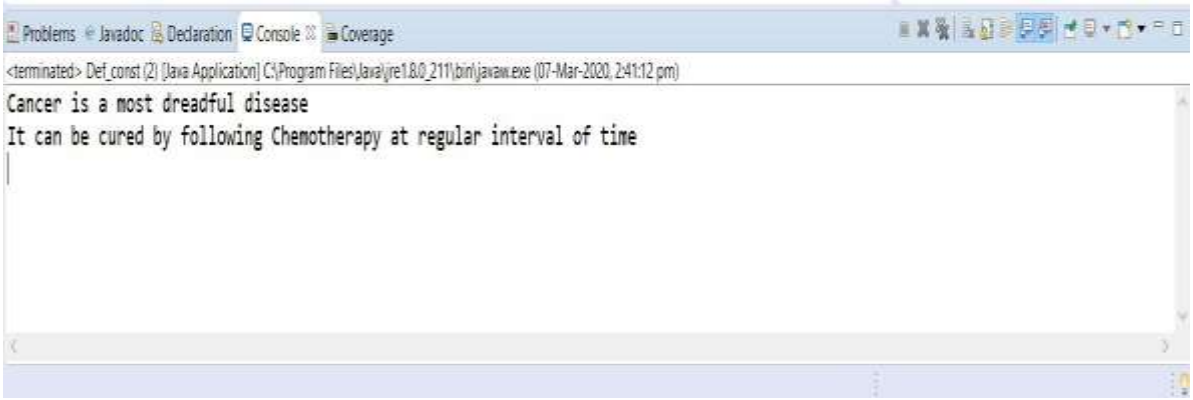
2. Use the default constructor in the program

3. End the program.

SOURCE CODE

```
package class_and_objects;
class Run{
    Run(){
        System.out.println("Cancer is a most dreadful disease");
    }
    void play(){
        String k="Chemotherapy";
        System.out.println("It can be cured by following "+k+" at regular interval
of time");
    }
}

public class Def_const
{

          public  void main(String[] args) {
                    Run r=new Run();
         r.play();

          }

}
```

OUTPUT



II COPY CONSTRUCTORS

ALGORITHM

1.Start the program.

2.Use the copy constructor in the program.

3.End the program.

SOURCE CODE

```
package class_and_objects;
class Patient{
    int id;
    String name;
    Patient(int i ,String n){
        id=i;
        name=n;
    }
      Patient(Patient s){
    id = s.id;
    name =s.name;
    }
     void display(){System.out.println(id+" "+name);
}
}

public class Copy_const
{

        public static void main(String[] args) {
                // TODO Auto-generated method stub
```

```
        Patient s1=new Patient(111,"Karthi");
        Patient s2=new Patient(s1);
        s1.display();
        s2.display();

    }
```

OUTPUT

III CONSTRUCTOR OVERLOADING

ALGORITHM

1. Start the program.
2. Use the constructor overloading in the program.
3. End the program.

SOURCE CODE

```
package class_and_objects;
class Animal
{
   Animal(){
      System.out.println("There are various kinds of animals");
   }
   Animal(String k){
      System.out.println("One variety is "+k+" which includes plants ");
   }
   Animal(String k,String g){
      System.out.println("The major species associated in this world is "+k+"
and the largest and furious predator is "+g);
   }
}
public class Const_over
{
```

```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Animal a=new Animal();
    Animal a1=new Animal("herbivores");
    Animal a2=new Animal("Carnivores","lion");


    }

}
```

OUTPUT:

RESULT

Thus the types of constructors are implemented and executed successfully.

EX.NO: 4 **JAVA INHERITANCE**

AIM

To work with different types of inheritances using java.

I SINGLE INHERITANCE

ALGORITHM

1. Start the program.
2. Implement the java single inheritance in the program.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.*;
//single inheritance
abstract class Species{
        String[] array = new String[20];
        int n;
        void get_animal_species()
        {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the no of species");
        n=in.nextInt();
        System.out.println("Enter the animal species");
        for(int i=0;i<n;i++)
        array[i]=in.next();
        }
        abstract void print_animals();
}
public class Humans extends Species {
        String conserve="Zoological parks";
         void print_animals() {
                for(int i=0;i<n;i++)
                System.out.println("The "+array[i]+" is the name of the species");

         }
         void print_results() {
                System.out.println("The "+conserve+" is responsible is
maintaining for this species");
        }

public static void main(String[] args) {
```

```
        Humans h=new Humans();
        Species s=h;
        s.get_animal_species();
        s.print_animals();
        h.print_results();



        }
}
```

OUTPUT

II MULTIPLE INHERITANCES

ALGORITHM

1.Start the program.

2.Implement the multiple inheritance in the program.

3.End the program.

SOURCE CODE:

```
package class_and_objects;
//multiple inheritance
interface Car
{
   int  speed=60;
   public void distanceTravelled();
}
interface Bus
{
   int distance=100;
   public void speed();
```

```
}
public class Vehicle  implements Car,Bus
{
   int distanceTravelled;
   int averageSpeed;
   public void distanceTravelled()
   {
      distanceTravelled=speed*distance;
      System.out.println("Total Distance Travelled is
:"+distanceTravelled);
   }
   public void speed()
   {
      int averageSpeed=distanceTravelled/speed;
      System.out.println("Average Speed maintained is : "+averageSpeed);
   }
   public static void main(String args[])
   {
      Vehicle v1=new Vehicle();
      v1.distanceTravelled();
      v1.speed();
   }
}
```

OUTPUT

III MULTILEVEL INHERITANCE

ALGORITHM

1.Start the program.

2.Use the multilevel inheritance in the program.

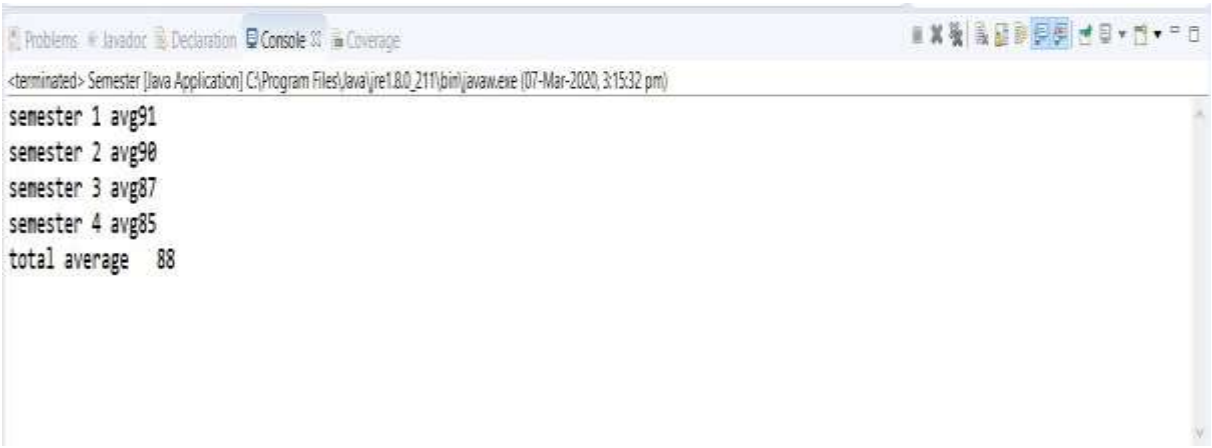
3. End the program.

SOURCE CODE

```
package class_and_objects;
//Multilevel inheritance
class Sem1
{
        int m11,m12,m13,m14,avg1;

        Sem1()
         {
            m11=95;
            m12=93;
            m13=88;
            m14=90;
             avg1=(m11+m12+m13+m14)/4;
         }
}
class Sem2 extends Sem1
{

         int m21,m22,m23,m24,avg2;
        Sem2()
        {
          m21=89;
          m22=98;
          m23=97;
          m24=79;
         avg2=(m21+m22+m23+m24)/4;
         }
}
class Sem3 extends Sem2
{

         int m31,m32,m33,m34,avg3;
         Sem3()
         {
           m31=87;
          m32=78;
          m33=88;
```

```
            m34=96;

                avg3=(m31+m32+m33+m34)/4;

          }

}

class Sem4 extends Sem3
{

   int m41,m42,m43,m44,avg4;
  Sem4()
 {
    m41=79;
    m42=90;
    m43=77;
    m44=96;avg4=(m41+m42+m43+m44)/4;
}
  int totatavg()
{
     return (avg1+avg2+avg3+avg4)/4;
}
}
class Semester
{
  public static void main(String args[])
  {

     Sem4 s4=new Sem4();
     System.out.println("semester 1 avg"+s4.avg1);
    System.out.println("semester 2 avg"+s4.avg2);
    System.out.println("semester 3 avg"+s4.avg3);
    System.out.println("semester 4 avg"+s4.avg4);

    System.out.println("total average   "+s4.totatavg());
  }
}
```

OUTPUT



IV HEIRARCHIAL INHERITANCE

ALGORITHM

1.Start the program.

2.Use the hierarchical inheritance in the program.

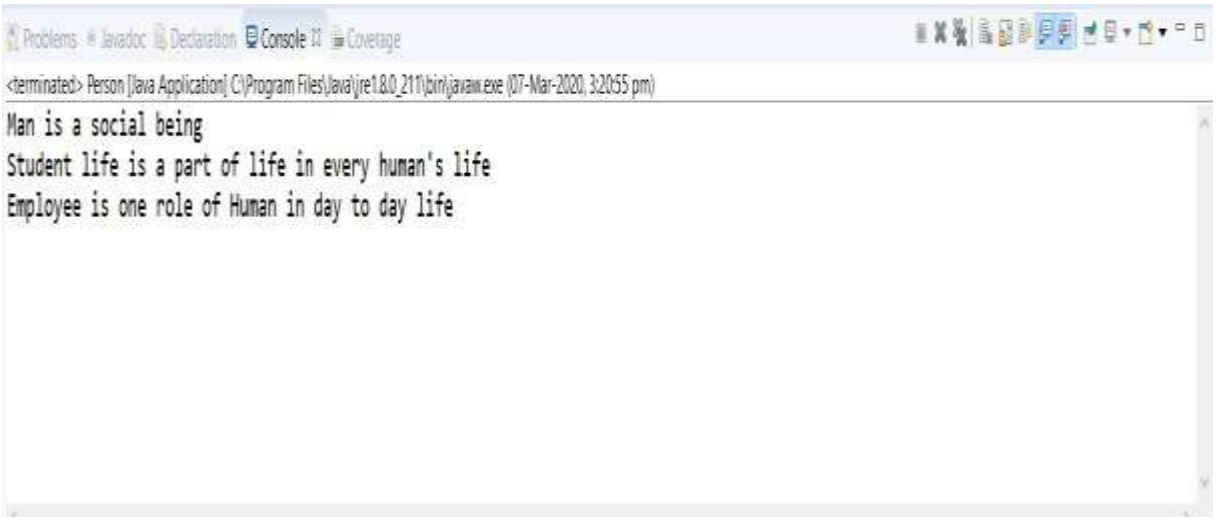
3.End the program.

SOURCE CODE:

```
 package class_and_objects;
//hierarchial Inheritance
class Human
{
   public void behaviour()
   {
      System.out.println("Man is a social being");
   }
}
class Employee extends Human
{
   public void role()
   {
      System.out.println("Employee is one role of Human in day to day life");
   }
}
class Student extends Human
{
  public void stage()
  {
     System.out.println("Student life is a part of life in every human's life");
  }
```

```
}

class Person
{
  public static void main(String args[])
  {
     Employee obj1 = new Employee();
     Student obj2 = new Student();
     //All classes can access the method of class A
     obj1.behaviour();
     obj2.stage();
     obj1.role();

  }
}
```

OUTPUT



V HYBRID INHERITANCE

ALGORITHM

1.Start the program.

2. Use the hybrid inheritance in the program.

3. End the program.

SOURCE CODE

```
package class_and_objects;
//Hybrid Inheritance

class Manoj
{
 String name="Manoj";
 void printX()
 {
 System.out.println(name);
 }

}

final class Football{
        final String nb="Messi";
        void func() {
        System.out.println(nb+" is the best player of all time");
}
}


class Play extends Manoj
{
 String b="Cricket";
 void print_play()
 {
 System.out.println(name+" plays "+b);
 }
}


class Cricket extends Play
{ int c=20;
 String sp="Virat Kohli";
 public void print_SS()
 {
 System.out.println(name+" has "+c+" years of experience in "+b);
 }

 void print_SBI()
 {
 print_SS();
 System.out.println(name+ "'s favourite player is "+sp);
 }
```

```
}

public class Sports
{
 public static void main(String[] args)
 {

  Cricket obj=new Cricket();
  Football f=new Football();

 obj.printX();
 obj.print_play();
 obj.print_SBI();
 f.func();

 }
}
```

OUTPUT

RESULT

Thus the types of inheritances are implemented and executed using java programs.

EX.NO: 5 JAVA POLYMORPHISM AND INTERFACES

AIM

To implement interfaces and polymorphism concepts in the java programs.

I MULTIPLE INHERITANCE USING INTERFACES

ALGORITHM

1.Start the program.

2.Use the interfaces to implement the multiple inheritance in the java program.

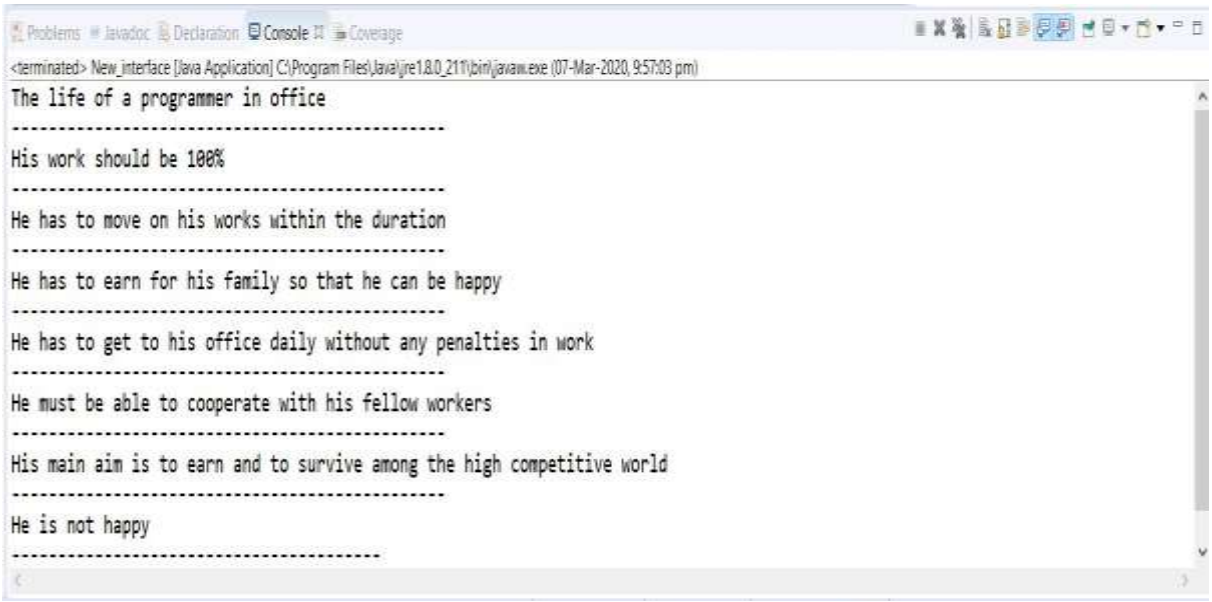
3.End the program.

SOURCE CODE

```
package class_and_objects;

interface Programmer{
	void program();
	void life();
	void office();
}

interface Family{
	void routine();
	void happiness();
}

class Personn implements Programmer,Family{
	public void program() {
		int b=100;
		System.out.println("The life of a programmer in office");
		System.out.println("-----------------------------------------");
		System.out.println("His work should be "+b+ "%");
		System.out.println("-----------------------------------------");
		System.out.println("He has to move on his works within the duration");
		System.out.println("-----------------------------------------");
	}
	public void life()
	{
		System.out.println("He has to earn for his family so that he can be happy");
		System.out.println("-----------------------------------------");
```

```
	}
	public void office() {
		System.out.println("He has to get to his office daily without any
penalties in work");
		System.out.println("-----------------------------------------");
		System.out.println("He must be able to cooperate with his fellow
workers");
		System.out.println("-----------------------------------------");
	}
	public void routine() {
		System.out.println("His main aim is to earn and to survive among
the high competitive world");
		System.out.println("-----------------------------------------");
	}
	public void happiness() {
		boolean b = false;
		if(b==true){
			System.out.println(b);
			System.out.println("He is happy");
			System.out.println("-------------------------------------");}
			else
			{
				System.out.println("He is not happy");
				System.out.println("-------------------------------------");
			}

	}

}
```

OUTPUT



II METHOD OVERRIDING

ALGORITHM

1.Start the program.

2.Implement method overriding through single inheritance.

3.End the program.

SOURCE CODE:

```
package class_and_objects;

class Company
{

	public void marketing()
	{
		int value=67;
		System.out.println("This is marketing sector of a company");
		System.out.println("The average sales management
"+value+"%");
		System.out.println("The company is planning to sale their
products at much lower prices");
	}
}


class eBay extends Company
{

	public void marketing()
	{
		System.out.println("eBay is one of the biggest sector of online
shopping");
		super.marketing();
	}
}

public class Over_rid {

	public static void main(String[] args) {
		eBay e=new eBay();
		e.marketing();
```

```
        }
}
```

OUTPUT

III METHOD OVERLOADING

ALGORITHM

1.Start the program.

2. Implement the argument-based method overloading.

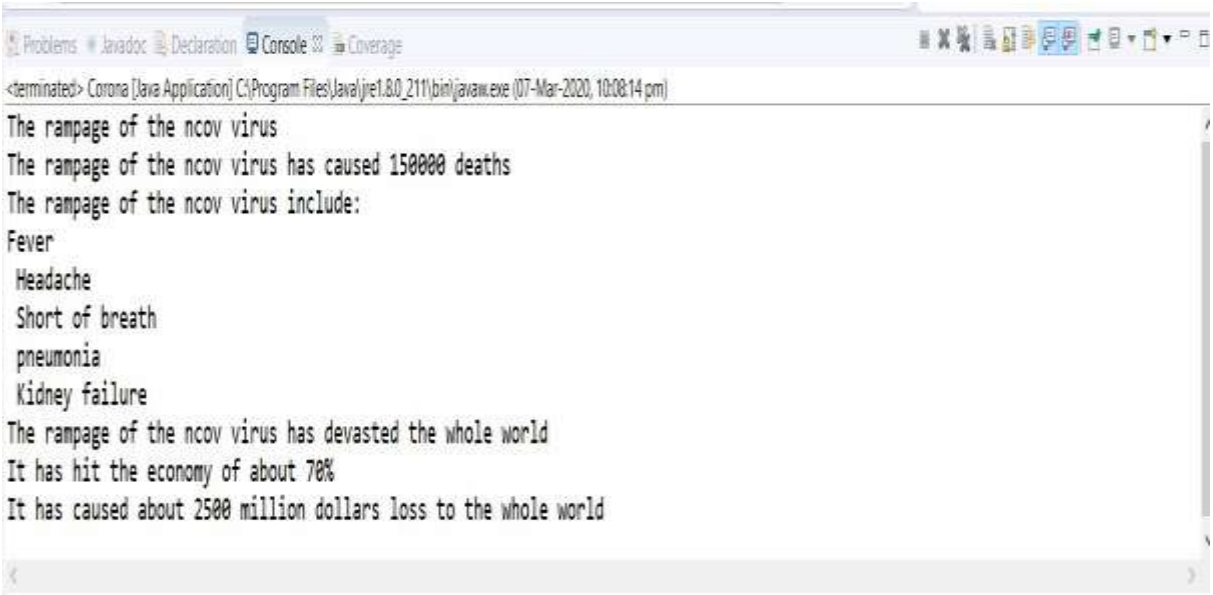
3. End the program.

SOURCE CODE

```
package class_and_objects;
class Virus
{
        void nCov(int n) {
                int numbers=n;
        System.out.println("The rampage of the ncov virus");
        System.out.println("The rampage of the ncov virus has caused
"+numbers+ " deaths");
        System.out.println("The rampage of the ncov virus include:");
        System.out.println("Fever\n Headache\n Short of breath\n pneumonia\n
Kidney failure");
        }
        void nCov(int e,int m) {
                int economy=e;
                int money=m;
```

```
                System.out.println("The rampage of the ncov virus has devasted
the whole world");
                System.out.println("It has hit the economy of about
"+economy+"%");
                System.out.println("It has caused about "+money+" million dollars
loss to the whole world");
                }
}

public class Corona {

        public static void main(String[] args) {
                Virus v=new Virus();
                v.nCov(150000);
                v.nCov(70, 2500);


        }

}
```

OUTPUT



RESULT

Thus, interfaces and polymorphism has been implemented using java programs

EX.NO: 6 **JAVA ARRAYS**

AIM

To implements arrays in programs with java.

I EQUALITY OF TWO ARRAYS

ALGORITHM

1.Start the program.

2.Input the number of elements of arr1 and arr2.

3.Input the elements of arr1 and arr2.

4.If all the elements of arr1 and arr2 are equal, then print "Same".

5.Else, print "Not Same".

6.End the program.

SOURCE CODE

```
package class_and_objects;
public class Array1
{
        static void  equality_checking_two_arrays(int[] my_array1, int[]
my_array2)
          {
          boolean  equalOrNot = true;

          if(my_array1.length == my_array2.length)
          {
          for (int i  = 0; i < my_array1.length; i++)
          {
          if(my_array1[i] != my_array2[i])
          {
          equalOrNot = false;}}}
          else
          {
          equalOrNot  = false;
          }

          if  (equalOrNot)
          {
          System.out.println("Two arrays are equal.");
          }
          else
```

```
        {
        System.out.println("Two  arrays are not equal.");}}

    public static void main(String[] args) {
            int[] array1 =  {2, 5, 7, 9, 11};
             int[] array2 =  {2, 5, 7, 8, 11};
             int[] array3 =  {2, 5, 7, 9, 11};
             equality_checking_two_arrays(array1,  array2);
             equality_checking_two_arrays(array1, array3);

    }

}
```

OUTPUT

II DELETION OF AN ELEMENT IN AN ARRAY

ALGORITHM

1: POS = POS - 1

TEMP = POS

2: Return A [POS]

3: Repeat Step4 while TEMP ≤ N-1

4: A [TEMP] = A [TEMP + 1]

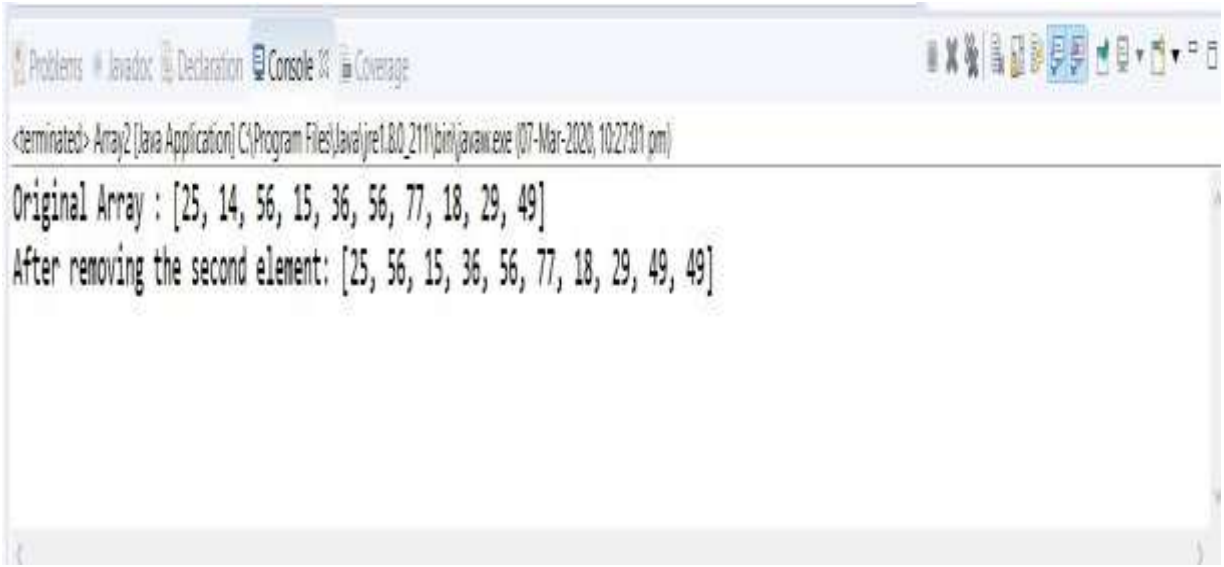
TEMP = TEMP + 1

5: N = N-1

SOURCE CODE

```
package class_and_objects;
import java.util.*;
public class Array2 {

	public static void main(String[] args) {
		int[] my_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

		System.out.println("Original Array :
"+Arrays.toString(my_array));

		// Remove the second element (index->1, value->14) of the
array
		int removeIndex = 1;

		for(int i = removeIndex; i < my_array.length -1; i++){
		    my_array[i] = my_array[i + 1];
		  }
		// We cannot alter the size of an array , after the removal, the last
and second last element in the array will exist twice
		System.out.println("After removing the second element:
"+Arrays.toString(my_array));

	}

}
```

OUTPUT



III INSERTION OF AN ELEMENT IN ARRAY

ALGORITHM

1: TEMP = N-1

POS = POS - 1

2: Repeat Step 3 While TEMP ≥ POS

3: A [TEMP+1] = A [TEMP]

TEMP = TEMP – 1

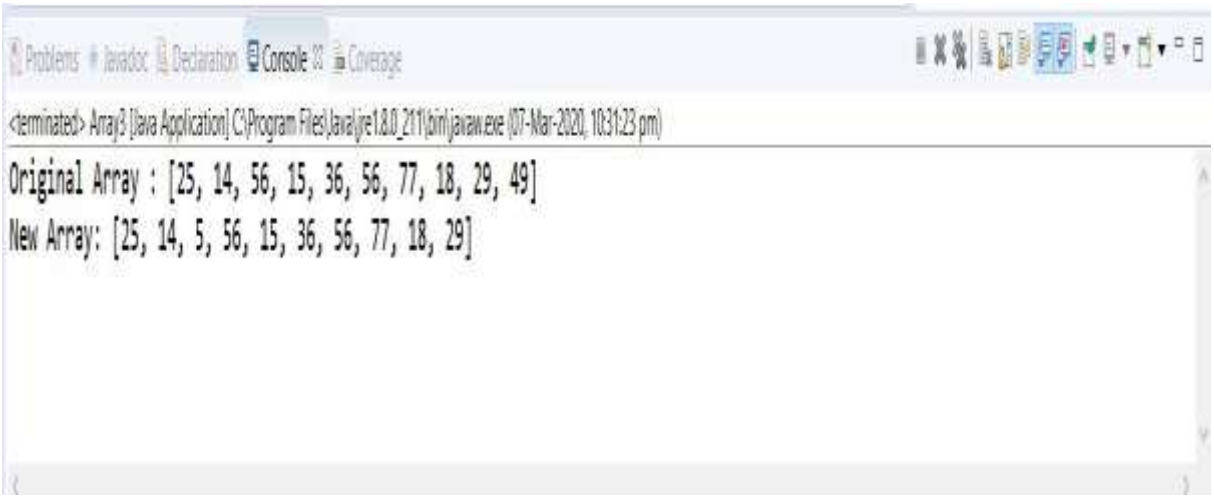
4: A [POS] = X

5: N = N + 1

SOURCE CODE

```
package class_and_objects;
import java.util.Arrays;
public class Array3
{

	public static void main(String[] args)
	{
		int[] my_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

	   // Insert an element in 3rd position of the array (index->2, value->5)

		int Index_position = 2;
		int newValue    = 5;

		System.out.println("Original Array : "+Arrays.toString(my_array));

		for(int i=my_array.length-1; i > Index_position; i--)
		{
			my_array[i] = my_array[i-1];
		}
		my_array[Index_position] = newValue;
		System.out.println("New Array: "+Arrays.toString(my_array));

	}

}
```

OUTPUT



IV PRINT UNIQUE VALUES FROM AN ARRAY

ALGORITHM

1.Declare and input the array elements.

2.Traverse the array from the beginning.

3.Check if the current element is found in the array again.

4.If it is found, then do not print that element.

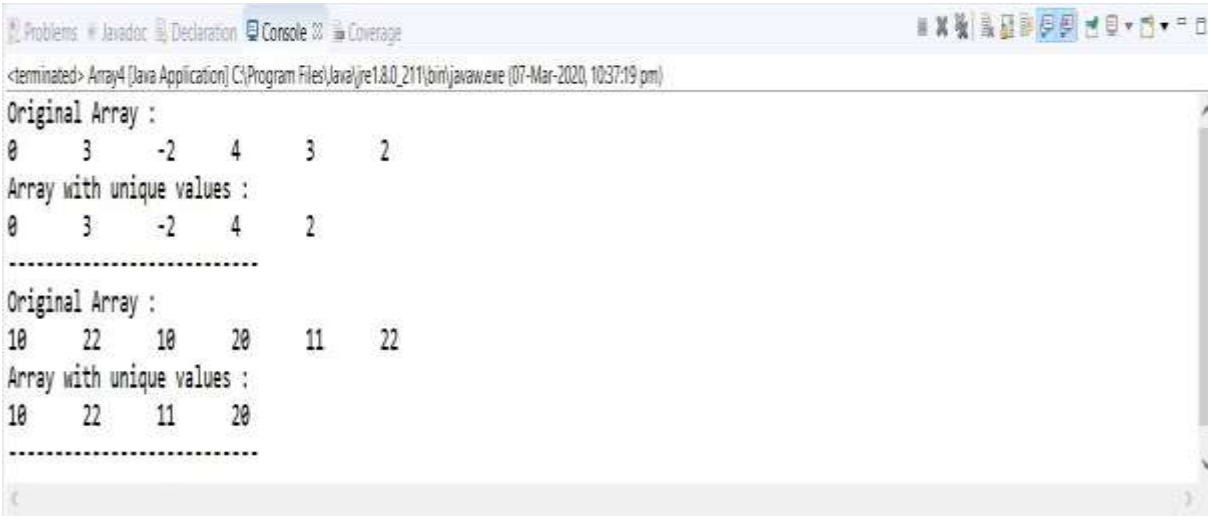
5.Else, print that element and continue.

SOURCE CODE

```
package class_and_objects;
import java.util.*;

public class Array4 {
        static void unique_array(int[] my_array)
   {
      System.out.println("Original Array : ");

      for (int i = 0; i < my_array.length; i++)
      {
         System.out.print(my_array[i]+"\t");
      }

     int no_unique_elements = my_array.length;

      for (int i = 0; i < no_unique_elements; i++)
      {
```

```
            for (int j = i+1; j < no_unique_elements; j++)
            {

                if(my_array[i] == my_array[j])
                {

                    my_array[j] = my_array[no_unique_elements-1];

                    no_unique_elements--;

                    j--;
                }
            }
        }



        int[] array1 = Arrays.copyOf(my_array, no_unique_elements);

        System.out.println();

        System.out.println("Array with unique values : ");

        for (int i = 0; i < array1.length; i++)
        {
            System.out.print(array1[i]+"\t");
        }

        System.out.println();

        System.out.println("--------------------------");
    }

        public static void main(String[] args) {

                unique_array(new int[] {0, 3, -2, 4, 3, 2});

               unique_array(new int[] {10, 22, 10, 20, 11, 22});
        }

}
```

OUTPUT



V JAGGED ARRAY

ALGORITHM

1.Start the program.

2.Implement the jagged array in the java program.

3.End the program.

SOURCE CODE

```
package class_and_objects;
public class JaggedArrays
{

        public static void main(String args[])
        {
        String student[][] = new String[3][];

        student[0] = new String[4];
        student[1] = new String[1];
        student[2] = new String[2];


        // 1st row
        student[0][0] = "Karthi";
        student[0][1] = "ROLL no:14";
        student[0][2] = "M1:21";
        student[0][3] = "M2:22";
        // 2nd row
        student[1][0] = "ROLL no:26";
        // 3rd row
```

```
student[2][0] = "Bhuvi";
student[2][1] = "Roll no:25";


System.out.println("Student Form");

for(int i = 0; i < student.length; i++)
{
for(int j = 0; j < student[i].length; j++)
{
System.out.print(student[i][j] + " ");
}
System.out.println();
}
}
}
```

OUTPUT

RESULT

Thus arrays are implemented using java programs.

EX.NO: 7 JAVA COLLECTIONS

AIM

To implement the various collections of the collection class in java programs.

I ARRAYLIST

ALGORITHM

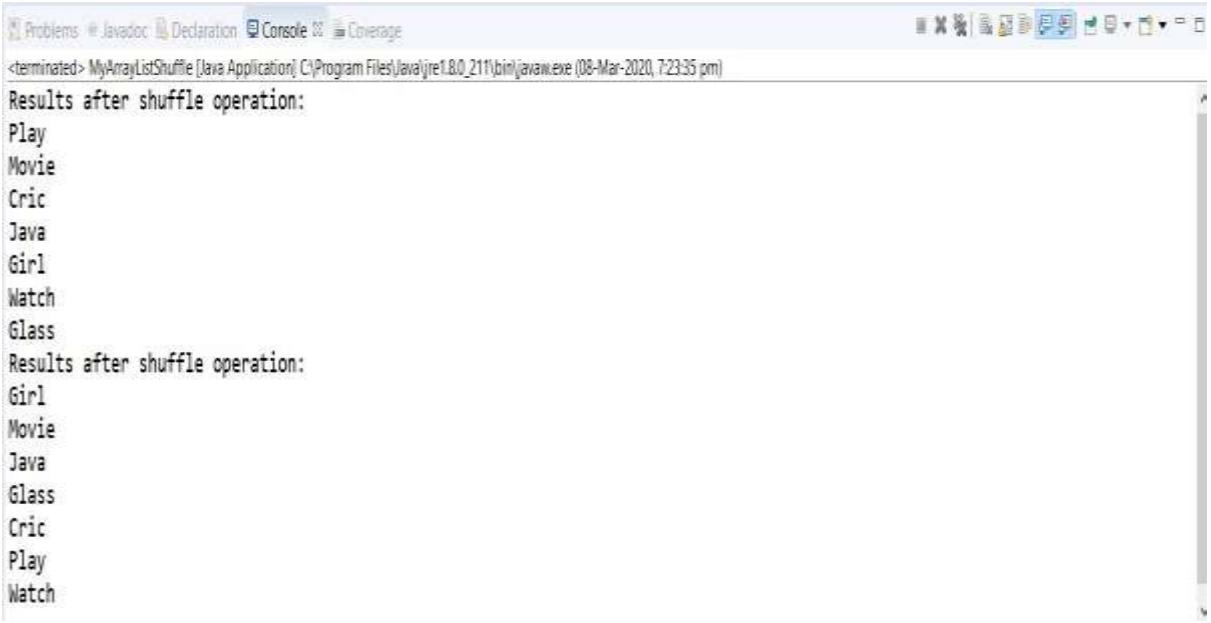
1. Start the program.
2. Implement the arraylist using generics.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.ArrayList;
import java.util.Collections;
public class MyArrayListShuffle
{
	public static void main(String[] args)
	{
		ArrayList<String> list = new ArrayList<String>();
		list.add("Java");
		list.add("Cric");
		list.add("Play");
		list.add("Watch");
		list.add("Glass");
		list.add("Movie");
		list.add("Girl");
		Collections.shuffle(list);
		System.out.println("Results after shuffle operation:");
		for(String str: list)
		 {
```

```
            System.out.println(str);
        }
        Collections.shuffle(list);
        System.out.println("Results after shuffle operation:");
        for(String str: list){
            System.out.println(str);

    }
    }
}
```

OUTPUT



II HASHSET

ALGORITHM

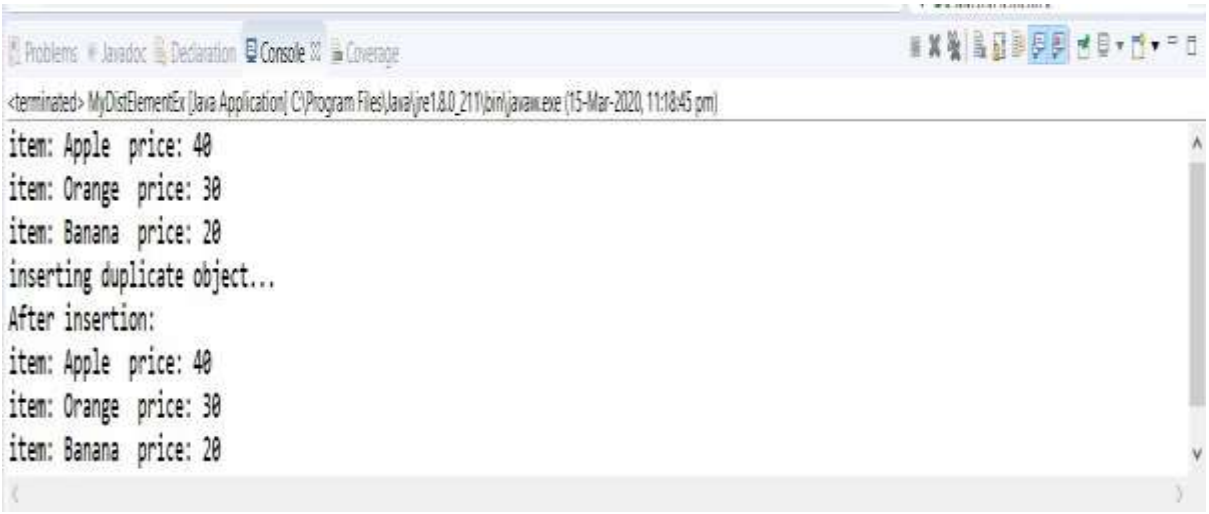
1. Start the program.
2. Implement hashset using generics.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.HashSet;
class Price
{

    private String item;
    private int price;

    public Price(String itm, int pr){
        this.item = itm;
        this.price = pr;
    }

    public int hashCode(){
        System.out.println("In hashcode");
        int hashcode = 0;
        hashcode = price*20;
        hashcode += item.hashCode();
        return hashcode;
    }

    public boolean equals(Object obj){
        System.out.println("In equals");
        if (obj instanceof Price) {
            Price pp = (Price) obj;
            return (pp.item.equals(this.item) && pp.price == this.price);
        } else {
            return false;
        }
    }

    public String getItem() {
        return item;
    }
    public void setItem(String item) {
        this.item = item;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }

    public String toString(){
```

```
        return "item: "+item+"  price: "+price;
    }
}
public class MyDistElementEx {

        public static void main(String[] args) {
                HashSet<Price> lhm = new HashSet<Price>();
        lhm.add(new Price("Banana", 20));
        lhm.add(new Price("Apple", 40));
        lhm.add(new Price("Orange", 30));
        for(Price pr:lhm){
            System.out.println(pr);
        }
        Price duplicate = new Price("Banana", 20);
        System.out.println("inserting duplicate object...");
        lhm.add(duplicate);
        System.out.println("After insertion:");
        for(Price pr:lhm){
            System.out.println(pr);

        }

}
}
```

OUTPUT



III HASHMAP

ALGORITHM

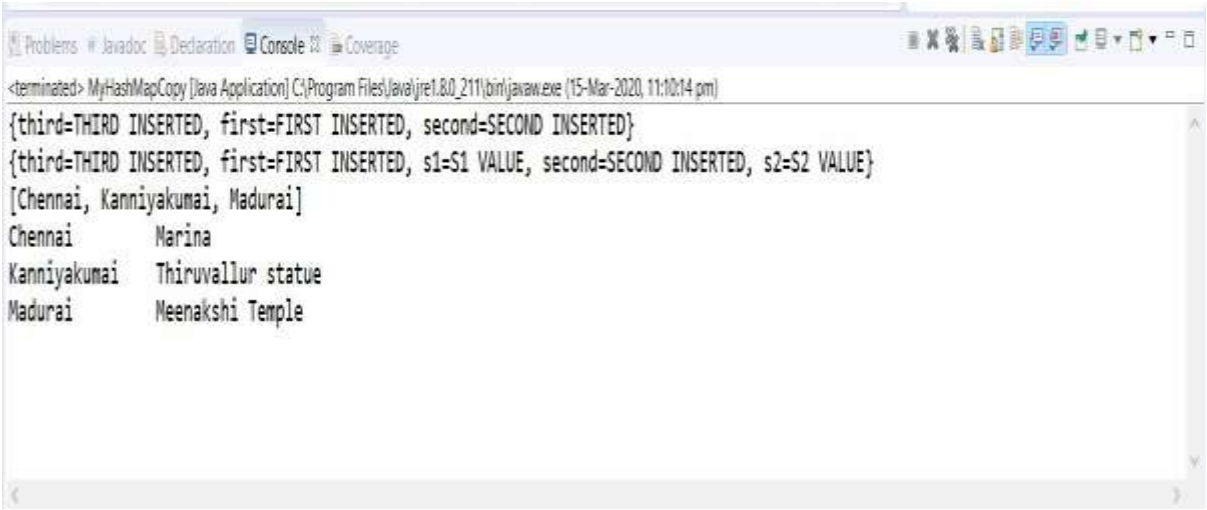
1. Start the program.
2. Implement the hashmap using generics.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Set;
public class MyHashMapCopy
{

        public static void main(String[] args) {
                HashMap<String, String> hm = new HashMap<String, String>();
            //add key-value pair to hashmap
            hm.put("first", "FIRST INSERTED");
            hm.put("second", "SECOND INSERTED");
            hm.put("third","THIRD INSERTED");
            System.out.println(hm);
            HashMap<String, String> subMap = new HashMap<String,
String>();
            subMap.put("s1", "S1 VALUE");
            subMap.put("s2", "S2 VALUE");
            hm.putAll(subMap);
            System.out.println(hm);
            HashMap<String, String> hm1 = new HashMap<String, String>();
            //add key-value pair to hashmap
            hm1.put("Chennai", "Marina");
```

```
            hm1.put("Madurai", "Meenakshi Temple");
            hm1.put("Kanniyakumai","Thiruvallur statue");
            System.out.println(hm1.keySet());
            Iterator ii=hm1.keySet().iterator();
            while(ii.hasNext()) {
              Object key=ii.next();
              Object value=hm1.get(key);
              System.out.print(key+"   \t");
              System.out.println(value);
            }
            }
}
```

OUTPUT



IV LINKED HASHMAP

ALGORITHM

1. Start the program.
2. Implement the linked hashmap using generics.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.LinkedHashMap;
import java.util.Set;
public class MyObjectKeySearch
{
public static void main(String a[])
{
        LinkedHashMap<Price_1, String> hm = new LinkedHashMap<Price_1,
String>();
        hm.put(new Price_1("Banana", 20), "Banana");
        hm.put(new Price_1("Apple", 40), "Apple");
        hm.put(new Price_1("Orange", 30), "Orange");
        printMap(hm);
        Price_1 key = new Price_1("Banana", 20);
        System.out.println("Does key available? "+hm.containsKey(key));
    }
    public static void printMap(LinkedHashMap<Price_1, String> map)
    {
        Set<Price_1> keys = map.keySet();
        for(Price_1 p:keys){
            System.out.println(p+"==>"+map.get(p));
        }
    }
}
class Price_1
{
    private String item;
    private int price;
```

```
public Price_1(String itm, int pr){
    this.item = itm;
    this.price = pr;
}

public int hashCode(){

    int hashcode = 0;
    hashcode = price*20;
    hashcode += item.hashCode();
    return hashcode;
}
public boolean equals(Object obj)
{
    if (obj instanceof Price_1) {
        Price_1 pp = (Price_1) obj;
        return (pp.item.equals(this.item) && pp.price == this.price);
    } else {
        return false;
    }
}
public String getItem() {
    return item;
}
public void setItem(String item) {
    this.item = item;
}
public int getPrice() {
```

```
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }

    public String toString(){
        return "item: "+item+"  price: "+price;
    }
}
```

OUTPUT

V HASHTABLE

ALGORITHM

1.Start the program.

2.Implement the hashtable using generics.

3.End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.Hashtable;
import java.util.Iterator;
class Emp{

    private String name;
    private int salary;
    private int id;

    public Emp(int id, String n, int s){
        this.id = id;
        this.name = n;
        this.salary = s;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getSalary() {
        return salary;
    }
    public void setSalary(int salary) {
        this.salary = salary;
    }
    public String toString(){
        return "Id: "+this.id+" -- Name: "+this.name+" -- Salary: "+this.salary;
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getId() {
        return id;
    }

    @Override
    public int hashCode() {
        return this.getId();
    }

    @Override
```

```
    public boolean equals(Object obj) {
        Emp e = null;
        if(obj instanceof Emp){
            e = (Emp) obj;
        }
        if(this.getId() == e.getId()){
            return true;
        } else {
            return false;
        }

    }
}
public class MyHashtableUserKeys
{

          public static void main(String[] args) {
                  Hashtable<Emp,String> tm = new Hashtable<Emp, String>();
        tm.put(new Emp(134,"Ram",3000), "RAM");
        tm.put(new Emp(235,"John",6000), "JOHN");
        tm.put(new Emp(876,"Crish",2000), "CRISH");
        tm.put(new Emp(512,"Tom",2400), "TOM");
        System.out.println("Fetching value by creating new key:");

        Emp e = new Emp(512,"Tom",2400);
        System.out.println(e+" ==> "+tm.get(e));
        Iterator ii=tm.keySet().iterator();
        while(ii.hasNext()) {
          Object key=ii.next();
          Object value=tm.get(key);
          System.out.print(key+"==>");
          System.out.println(value);
        }

          }

}
```

OUTPUT

VI LINKED HASHSET

ALGORITHM

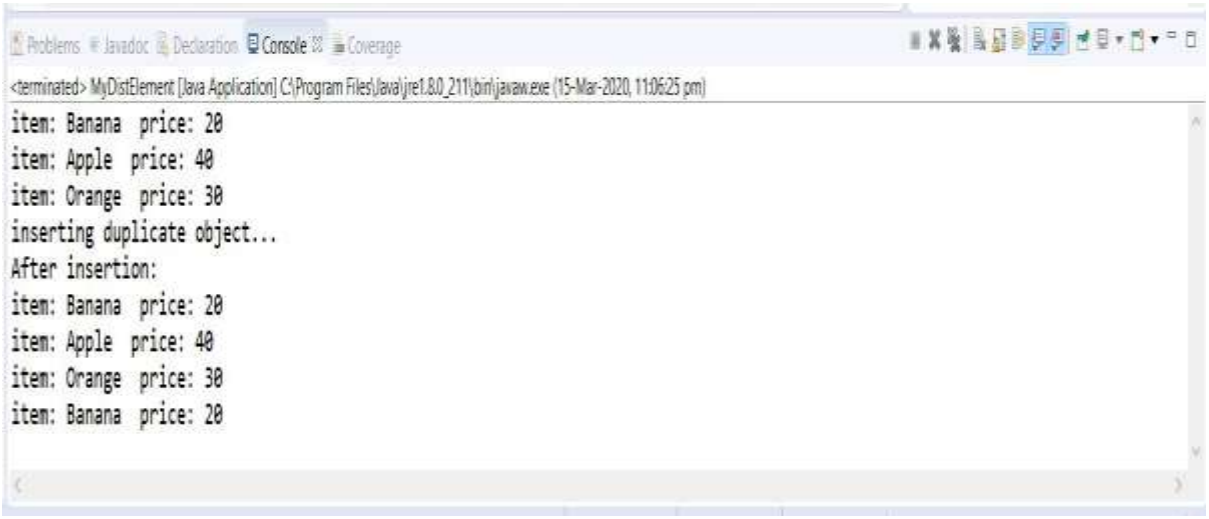
1. Start the program.
2. Implement the linked hashset using generics.
3. End the program.

SOURCE CODE:

```
package class_and_objects;
import java.util.LinkedHashSet;
class Amount{

    private String item;
    private int price;

    public Amount(String itm, int pr){
        this.item = itm;
        this.price = pr;
    }

    public int hashCode(){
        int hashcode = 0;
        hashcode = price*20;
        hashcode += item.hashCode();
        return hashcode;
    }

    public boolean equals(Object obj){
        if (obj instanceof Price) {
            Amount pp = (Amount) obj;
```

```
            return (pp.item.equals(this.item) && pp.price == this.price);
        } else {
            return false;
        }
    }

    public String getItem() {
        return item;
    }
    public void setItem(String item) {
        this.item = item;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }

    public String toString(){
        return "item: "+item+"  price: "+price;
    }
}
public class MyDistElement {

          public static void main(String[] args) {
                  LinkedHashSet<Amount> lhm = new LinkedHashSet<Amount>();
        lhm.add(new Amount("Banana", 20));
        lhm.add(new Amount("Apple", 40));
        lhm.add(new Amount("Orange", 30));
        for(Amount pr:lhm){
            System.out.println(pr);
        }
        Amount duplicate = new Amount("Banana", 20);
        System.out.println("inserting duplicate object...");
        lhm.add(duplicate);
        System.out.println("After insertion:");
        for(Amount pr:lhm){
            System.out.println(pr);

          }

}
}
```

OUTPUT



VII TREEMAP

ALGORITHM

1. Start the program.
2. Implement the treemap using generics.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.Map.Entry;
import java.util.*;

public class MyTreeMapFirstElement {

    public static void main(String a[]){
        //By using salary comparator (int comparison)
        TreeMap<Emp1,String> trmap = new TreeMap<Emp1, String>(new
MySalaryCompr());
        trmap.put(new Emp1("Ram",3000), "RAM");
        trmap.put(new Emp1("John",6000), "JOHN");
        trmap.put(new Emp1("Crish",2000), "CRISH");
```

```
        trmap.put(new Emp1("Tom",2400), "TOM");
        Emp1 em = trmap.firstKey();
        Iterator ii=trmap.keySet().iterator();
        while(ii.hasNext()) {
          Object key=ii.next();
          Object value=trmap.get(key);
          System.out.print(key+"   \t\n");
        }
        System.out.println("Highest salary emp: "+em);
    }
}
class MySalaryCompr implements Comparator<Emp1>
{
    @Override
    public int compare(Emp1 e1, Emp1 e2) {
        if(e1.getSalary() < e2.getSalary()){
            return 1;
        } else {
            return -1;
        }
    }
}
class Emp1
{
    private String name;
    private int salary;

    public Emp1(String n, int s){
        this.name = n;
```

```
        this.salary = s;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getSalary() {
        return salary;
    }
    public void setSalary(int salary) {
        this.salary = salary;
    }
    public String toString(){
        return "Name: "+this.name+"-- Salary: "+this.salary;
    }
}
```

OUTPUT

VIII VECTOR

ALGORITHM

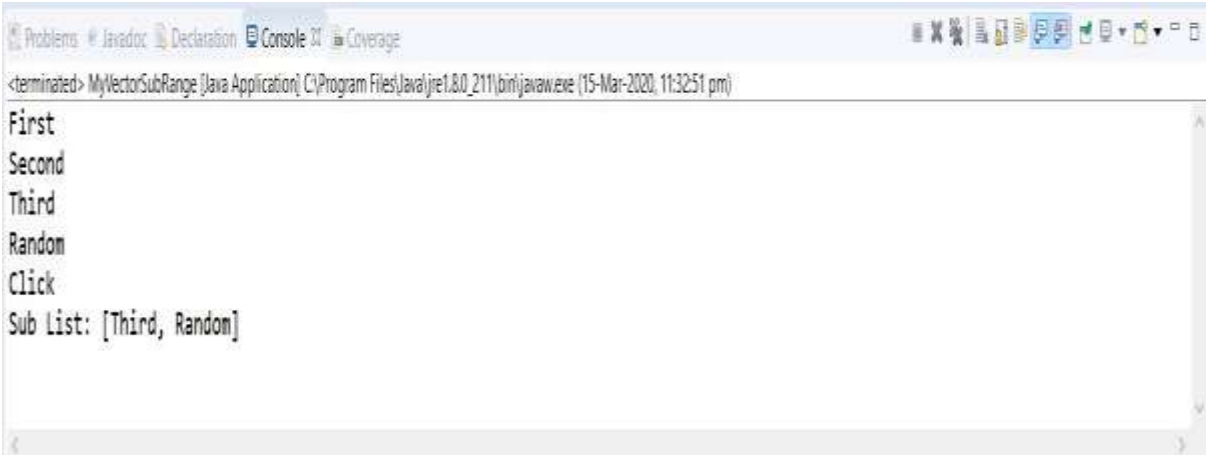
1. Start the program.
2. Implement the vector using generics.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.Iterator;
import java.util.List;
import java.util.Vector;
import java.util.Map.Entry;
public class MyVectorSubRange
{
        public static void main(String[] args) {
                Vector<String> vct = new Vector<String>();
      //adding elements to the end
      vct.add("First");
      vct.add("Second");
      vct.add("Third");
      vct.add("Random");
      vct.add("Click");

      Iterator ii=vct.iterator();
      while(ii.hasNext()) {
        Object key=ii.next();
        System.out.print(key+"   \t\n");
      }
      List<String> list = vct.subList(2, 4);
      System.out.println("Sub List: "+list);
```

```
        }
}
```

OUTPUT



IX TREESET

ALGORITHM

1.Start the program.

2.Implement the treeset using generics.

3.End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.Comparator;
import java.util.Iterator;
import java.util.TreeSet;
class MySalCompr1 implements Comparator<Empl2>{

    @Override
    public int compare(Empl2 e1, Empl2 e2) {
        if(e1.getSalary() > e2.getSalary()){
            return 1;
        } else {
            return -1;
        }
```

```
    }
}
class Empl2
{
    private String name;
    private int salary;

    public Empl2(String n, int s){
        this.name = n;
        this.salary = s;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getSalary() {
        return salary;
    }
    public void setSalary(int salary) {
        this.salary = salary;
    }
    public String toString(){
        return "Name: "+this.name+"-- Salary: "+this.salary;
    }
}
```

```
public class MyHighSalEmp
{
        public static void main(String[] args) {
                TreeSet<Empl2> salComp = new TreeSet<Empl2>(new
MySalCompr1());
      salComp.add(new Empl2("Ram",3000));
      salComp.add(new Empl2("John",6000));
      salComp.add(new Empl2("Crish",2000));
      salComp.add(new Empl2("Tom",2400));
      Iterator ii=salComp.iterator();
      while(ii.hasNext()) {
        Object key=ii.next();
        System.out.print(key+"    \t\n");
      }
      System.out.println("Highest salary emp: "+salComp.last());
        }
}
```

OUTPUT

X LINKEDLIST

ALGORITHM

1. Start the program.
2. Implement the linkedlist using generics.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.Collections;
import java.util.Comparator;
import java.util.LinkedList;
class MySalaryComp implements Comparator<Empl>{

    @Override
    public int compare(Empl e1, Empl e2) {
        if(e1.getSalary() < e2.getSalary()){
            return 1;
        } else {
            return -1;
        }
    }
}
class Empl{

    private String name;
    private int salary;

    public Empl(String n, int s){
        this.name = n;
        this.salary = s;
```

```
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getSalary() {
        return salary;
    }
    public void setSalary(int salary) {
        this.salary = salary;
    }
    public String toString(){
        return "Name: "+this.name+"-- Salary: "+this.salary;
    }
}
public class MyLinkedListSort {

          public static void main(String[] args) {
                  LinkedList<Empl> list = new LinkedList<Empl>();
        list.add(new Empl("Ram",3000));
        list.add(new Empl("John",6000));
        list.add(new Empl("Crish",2000));
        list.add(new Empl("Tom",2400));
        Collections.sort(list,new MySalaryComp());
        System.out.println("Sorted list entries: ");
        for(Empl e:list){
```

```
        System.out.println(e);
}
}
}
```

OUTPUT

RESULT

Thus the types of collections are implemented using java programs.

EX.NO: 8 **JAVA SWINGS GUI**

AIM

To implement swing using java.

I CALCULATOR

ALGORITHM

1. Start the program.
2. Use jlabel, jbutton, jtextfield in the program.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;

public class Swing1 extends javax.swing.JFrame implements ActionListener {
static JFrame f;

    // create a textfield
    static JTextField l;

    // store oprerator and operands
    String s0, s1, s2;

    // default constrcutor
    public Swing1()
    {
        s0 = s1 = s2 = "";
    }
```

```
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 888, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 550, Short.MAX_VALUE)
        );


        pack();
    }// </editor-fold>


    public static void main(String args[]) {
        f = new JFrame("Swing1");


        try {
```

```
            // set look and feel

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName()
);
        }
        catch (Exception e) {
            System.err.println(e.getMessage());
        }

        // create a object of class
        Swing1 c = new Swing1();

        // create a textfield
        l = new JTextField(16);

        // set the textfield to non editable
        l.setEditable(false);

        // create number buttons and some operators
        JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, ba, bs, bd, bm, be, beq,
beq1;

        // create number buttons
        b0 = new JButton("0");
        b1 = new JButton("1");
        b2 = new JButton("2");
        b3 = new JButton("3");
        b4 = new JButton("4");
        b5 = new JButton("5");
        b6 = new JButton("6");
```

```
b7 = new JButton("7");
b8 = new JButton("8");
b9 = new JButton("9");

// equals button
beq1 = new JButton("=");

// create operator buttons
ba = new JButton("+");
bs = new JButton("-");
bd = new JButton("/");
bm = new JButton("*");
beq = new JButton("C");

// create . button
be = new JButton(".");

// create a panel
JPanel p = new JPanel();

// add action listeners
bm.addActionListener(c);
bd.addActionListener(c);
bs.addActionListener(c);
ba.addActionListener(c);
b9.addActionListener(c);
b8.addActionListener(c);
b7.addActionListener(c);
b6.addActionListener(c);
```

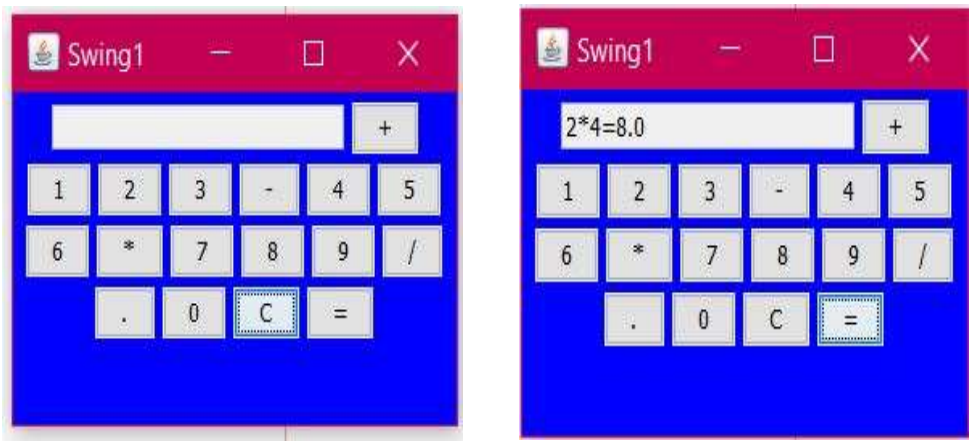
```
b5.addActionListener(c);
b4.addActionListener(c);
b3.addActionListener(c);
b2.addActionListener(c);
b1.addActionListener(c);
b0.addActionListener(c);
be.addActionListener(c);
beq.addActionListener(c);
beq1.addActionListener(c);

// add elements to panel
p.add(l);
p.add(ba);
p.add(b1);
p.add(b2);
p.add(b3);
p.add(bs);
p.add(b4);
p.add(b5);
p.add(b6);
p.add(bm);
p.add(b7);
p.add(b8);
p.add(b9);
p.add(bd);
p.add(be);
p.add(b0);
p.add(beq);
p.add(beq1);
```

```
    // set Background of panel
    p.setBackground(Color.blue);

    // add panel to frame
    f.add(p);

    f.setSize(200, 220);
    f.show();
}
public void actionPerformed(ActionEvent e)
{
    String s = e.getActionCommand();

    // if the value is a number
    if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') || s.charAt(0) == '.') {
        // if operand is present then add to second no
        if (!s1.equals(""))
            s2 = s2 + s;
        else
            s0 = s0 + s;

        // set the value of text
        l.setText(s0 + s1 + s2);
    }
    else if (s.charAt(0) == 'C') {
        // clear the one letter
        s0 = s1 = s2 = "";
```

```
        // set the value of text
        l.setText(s0 + s1 + s2);
    }
    else if (s.charAt(0) == '=') {

        double te;

        // store the value in 1st
        if (s1.equals("+"))
            te = (Double.parseDouble(s0) + Double.parseDouble(s2));
        else if (s1.equals("-"))
            te = (Double.parseDouble(s0) - Double.parseDouble(s2));
        else if (s1.equals("/"))
            te = (Double.parseDouble(s0) / Double.parseDouble(s2));
        else
            te = (Double.parseDouble(s0) * Double.parseDouble(s2));

        // set the value of text
        l.setText(s0 + s1 + s2 + "=" + te);

        // convert it to string
        s0 = Double.toString(te);

        s1 = s2 = "";
    }
    else {
        // if there was no operand
        if (s1.equals("") || s2.equals(""))
            s1 = s;
```

```
            // else evaluate
            else {
                double te;

                // store the value in 1st
                if (s1.equals("+"))
                    te = (Double.parseDouble(s0) + Double.parseDouble(s2));
                else if (s1.equals("-"))
                    te = (Double.parseDouble(s0) - Double.parseDouble(s2));
                else if (s1.equals("/"))
                    te = (Double.parseDouble(s0) / Double.parseDouble(s2));
                else
                    te = (Double.parseDouble(s0) * Double.parseDouble(s2));

                // convert it to string
                s0 = Double.toString(te);

                // place the operator
                s1 = s;

                // make the operand blank
                s2 = "";
            }

            // set the value of text
            l.setText(s0 + s1 + s2);
        }
    }
}
```

OUTPUT



II CHARACTER FUNCTIONS

ALGORITHM

1. Start the program.
2. Use jlabel, jbutton, jtextfield in the program.
3. End the program.

SOURCE CODE

```
public class Srevchar extends javax.swing.JFrame {
    /**
     * Creates new form Srevchar
     */
    public Srevchar() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
```

```
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jTextField1 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jButton1.setText("reverse");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("space character");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jButton3.setText("Vowels/consonants");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton3ActionPerformed(evt);
        }
    });

    jLabel1.setText("Enter");

    jTextField2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextField2ActionPerformed(evt);
        }
    });

    jLabel2.setText("Result");

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)
                .addComponent(jLabel1)
                .addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jButton1)
                        .addGap(44, 44, 44)
                        .addComponent(jButton2))
                    .addComponent(jTextField1))
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 218, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(20, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton3)
            .addGap(74, 74, 74))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(37, 37, 37)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jButton1)
                    .addComponent(jButton2))
                .addGap(18, 18, 18)
                .addComponent(jButton3)
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2))
                .addContainerGap(37, Short.MAX_VALUE))
        );


        pack();
    }// </editor-fold>


    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }


    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String text=jTextField1.getText();
```

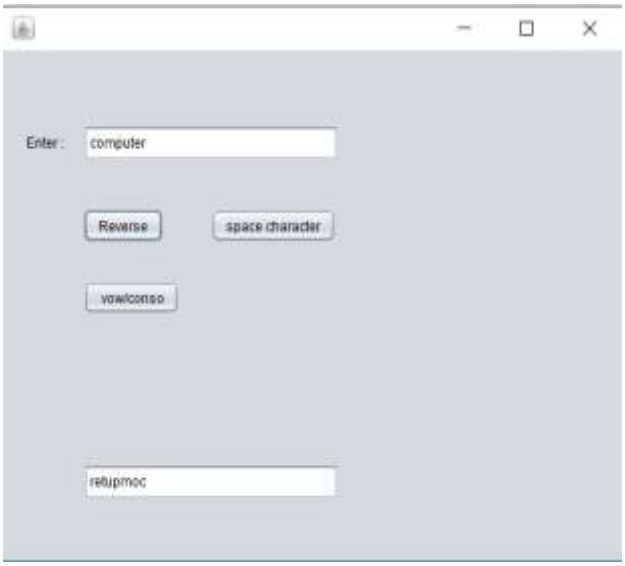
```
        StringBuilder rev=new StringBuilder(text);
        jTextField2.setText(""+rev.reverse());

    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        String word= jTextField1.getText();
int count=0;
char[] txt=new char[word.length()];
 txt=word.toCharArray();
 for(int i=0;i<word.length();i++)
 {
      if(' '== txt[i])
          count++;
 }
jTextField2.setText("No. of space character:"+count);

    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        String word= jTextField1.getText();
int vcount=0,ccount=0;
char[] txt=new char[word.length()];
 txt=word.toCharArray();
 for(int i=0;i<word.length();i++)
 {
      if('a'== txt[i]||'u'== txt[i]||'o'== txt[i]||'i'== txt[i]||'e'== txt[i]||'A'== txt[i]||'U'==
txt[i]||'O'== txt[i]||'I'== txt[i]||'E'== txt[i])
          vcount++;
```

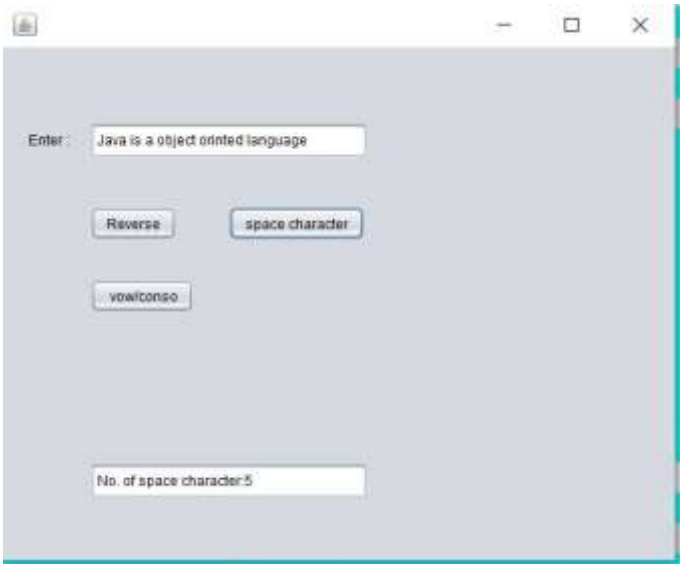
```
        else
            ccount++;
    }
jTextField2.setText("No. of vowels:"+vcount+"\n   No.of Consonants"+ccount)}


    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(Srevchar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

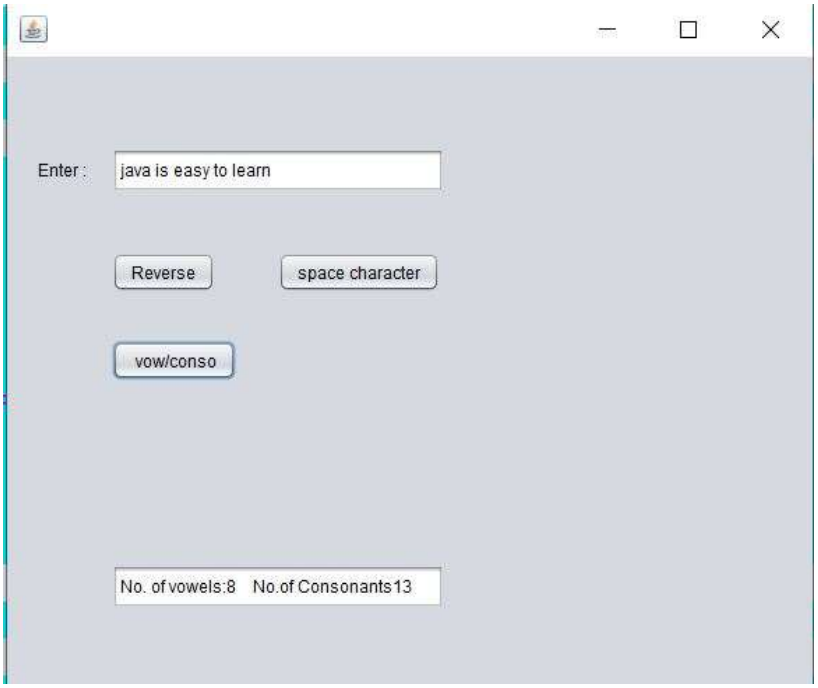
```
java.util.logging.Logger.getLogger(Srevchar.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Srevchar.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Srevchar.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        }
        //</editor-fold>


        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Srevchar().setVisible(true);
            }
        };
    }


    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration
```

}

OUTPUT:


Enter :
computer
Reverse
space character
vow/conso
retupmoc


Enter :
Java is a object orinted language
Reverse
space character
vow/conso
No. of space character:5


Enter :
java is easy to learn
Reverse
space character
vow/conso
No. of vowels:8 No.of Consonants13

III OPERATIONS ON NUMBER

ALGORITHM

1. Start the program.
2. Use jbutton, jtextfield, jtextarea,jlabel,jbuttons in the program .
3. End the program.

SOURCE CODE

```
public class aaa extends javax.swing.JFrame {


    public aaa() {
        initComponents();
    }


    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {


        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTextArea2 = new javax.swing.JTextArea();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
```

```
        jButton3 = new javax.swing.JButton();
        jButton4 = new javax.swing.JButton();
        jButton5 = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jTextArea1.setColumns(20);
        jTextArea1.setRows(5);
        jTextArea1.setText("ENTER A NUMBER HERE");
        jScrollPane1.setViewportView(jTextArea1);

        jTextArea2.setColumns(20);
        jTextArea2.setRows(5);
        jScrollPane2.setViewportView(jTextArea2);

        jButton1.setText("ARMSTRONG");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("PALINDROME");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jButton3.setText("SUM OF DIGITS");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton3ActionPerformed(evt);
        }
    });

    jButton4.setText("REVERSE");
    jButton4.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton4ActionPerformed(evt);
        }
    });

    jButton5.setText("PRIME");
    jButton5.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton5ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(layout.createSequentialGroup()
```

```
                .addGap(58, 58, 58)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(70, 70, 70)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jButton5)
                    .addComponent(jButton2)
                    .addComponent(jButton1)
                    .addComponent(jButton3)
                    .addComponent(jButton4)))
            .addGroup(layout.createSequentialGroup()
                .addGap(28, 28, 28)
                .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 507,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(316, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(28, 28, 28)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jButton1)
                    .addGap(23, 23, 23)
                    .addComponent(jButton2)
                    .addGap(18, 18, 18)
                    .addComponent(jButton3)
                    .addGap(18, 18, 18)
                    .addComponent(jButton4)))
            .addGap(18, 18, 18)
            .addComponent(jButton5)
            .addGap(18, 18, 18)
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(157, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         String a1 = jTextArea1.getText();
         int a = Integer.parseInt(a1);
         int sum = 0;
          while(a!=0)
          {
              sum = sum + a%10;
              a = a/10;
```

```
        }
        jTextArea2.setText("\nSUM OF DIGITS = "+ sum);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String a1 = jTextArea1.getText();
        int a = Integer.parseInt(a1);
        int temp,sum,r;
         sum = 0;
         temp = a;
         while(a>0)
         {
            r = a%10;
            sum = sum + (r*r*r);
            a = a/10;
         }
         if(sum == temp)
            jTextArea2.setText("IT IS AN ARMSTRONG NUMBER");
         else
             jTextArea2.setText("IT IS NOT AN ARMSTRONG NUMBER");
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String a1 = jTextArea1.getText();
         int a = Integer.parseInt(a1);
         int temp = a;
          int rev = 0;
```

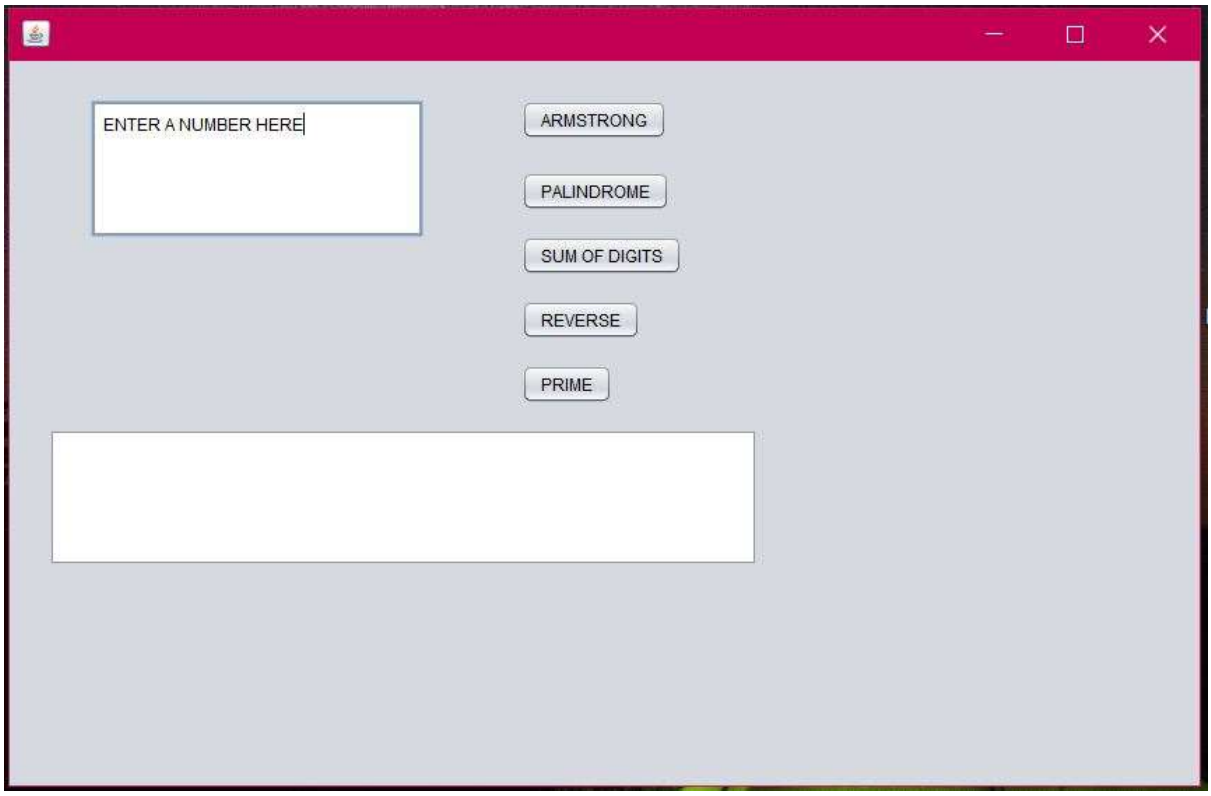
```
        while(a != 0)
        {
            rev = (rev*10) + a%10;
            a = a/10;
        }
        if(temp == rev)
            jTextArea2.setText("\nIT IS A PALINDROME NUMBER");
        else
            jTextArea2.setText("\nIT IS NOT A PALINDROME NUMBER");
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String a1 = jTextArea1.getText();
        int a = Integer.parseInt(a1);
        int rev = 0;
        while(a != 0)
        {
            rev = (rev*10) + a%10;
            a = a/10;
        }
        jTextArea2.setText("\nREVERSE = "+ rev);
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String a1 = jTextArea1.getText();
        int a = Integer.parseInt(a1);
        int f = 1;
```

```
            if(a==4)
               f=0;
            else
            {
               for(int i=2;i<a/2;i++)
               {
                  if(a%i==0)
                  {
                     f=0;
                     break;
                  }
               }
            }
            if(f==0)
               jTextArea2.setText("\nNOT PRIME");
            else
               jTextArea2.setText("\nPRIME");
   }

   /**
    * @param args the command line arguments
    */
   public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
```

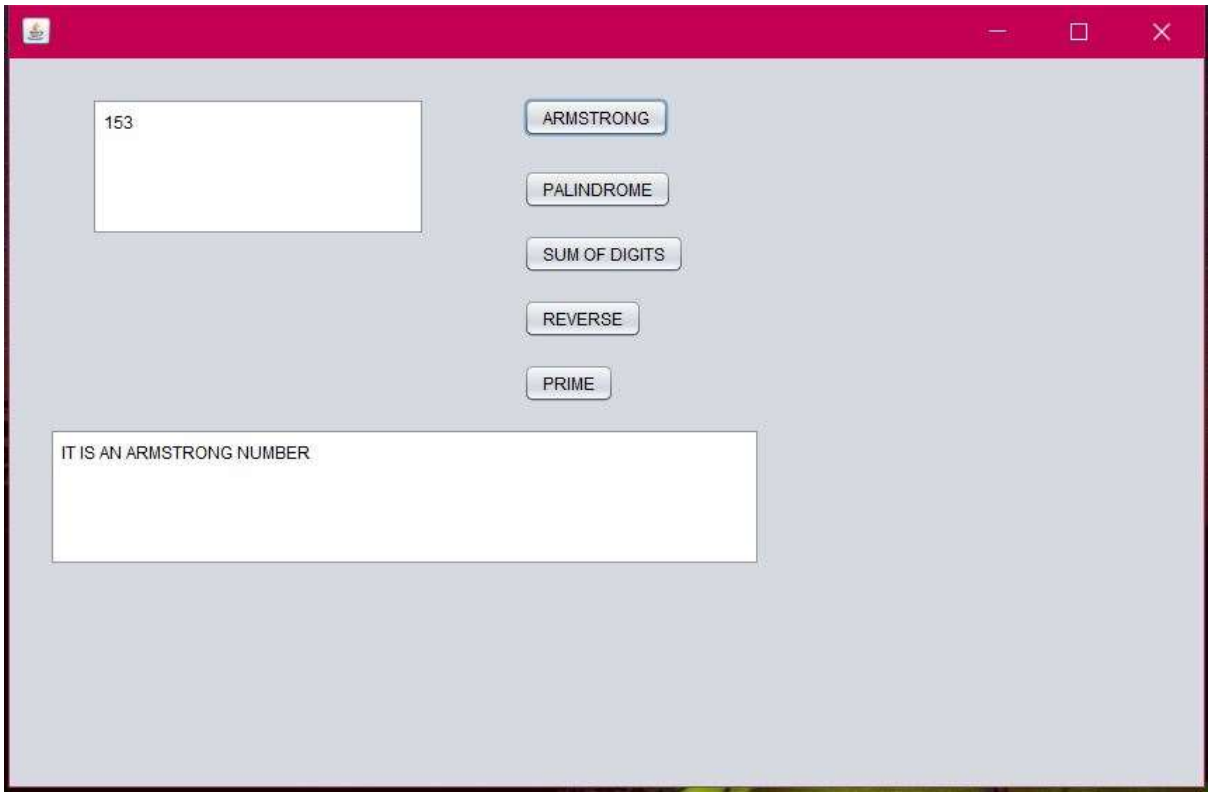
```
        */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(aaa.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(aaa.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(aaa.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(aaa.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new aaa().setVisible(true);
```

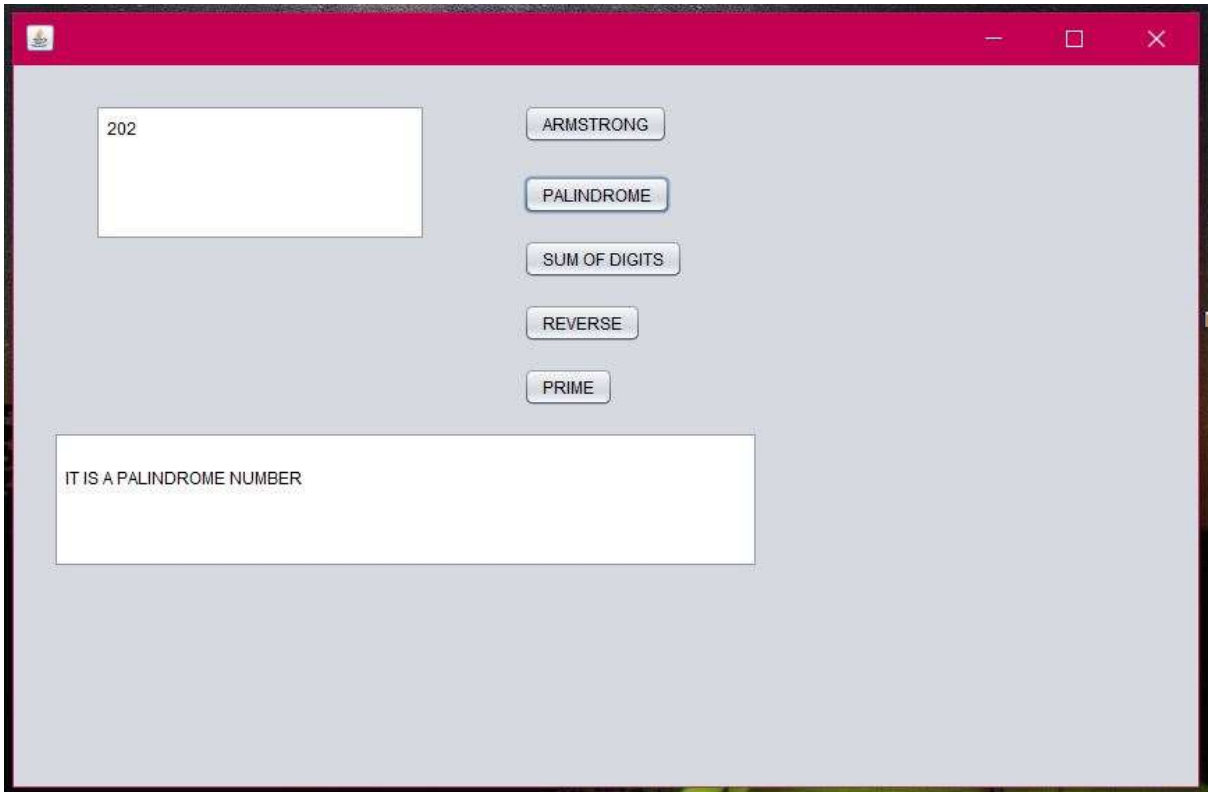
```
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextArea jTextArea2;
// End of variables declaration
}
```

OUTPUT

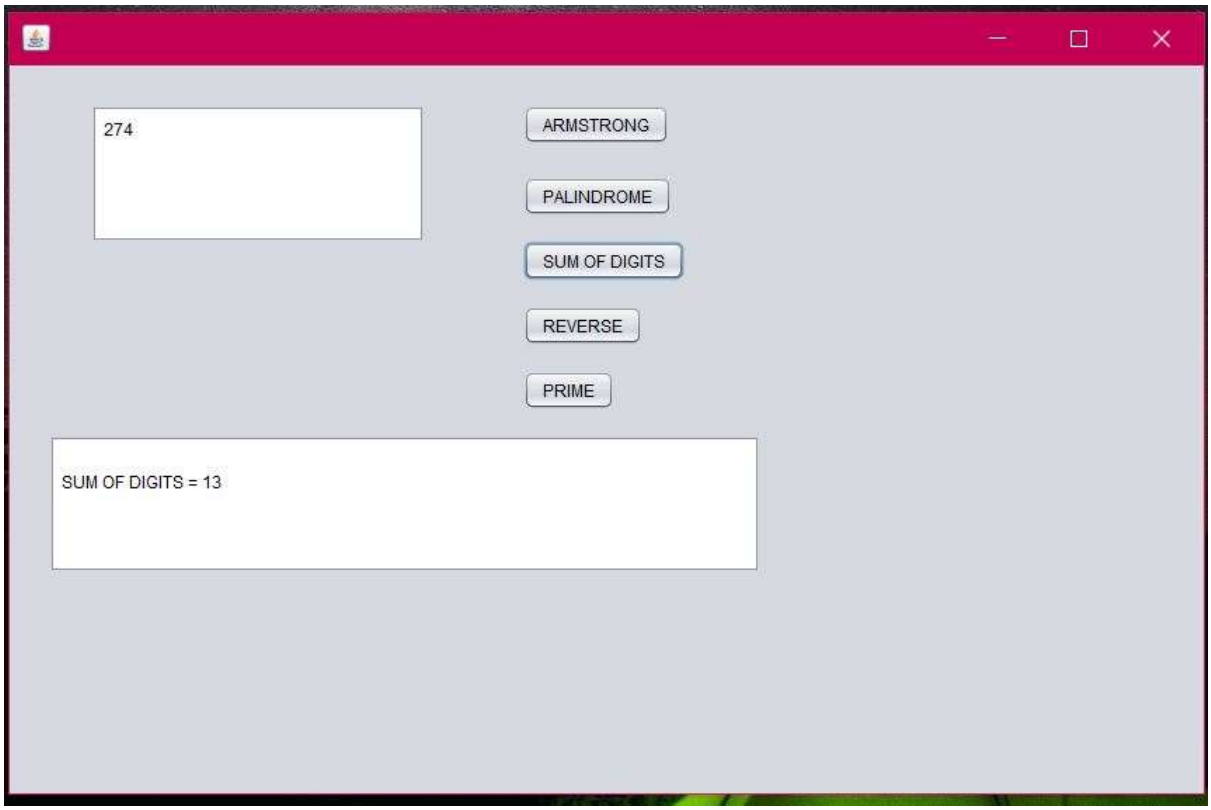

ENTER A NUMBER HERE
ARMSTRONG
PALINDROME
SUM OF DIGITS
REVERSE
PRIME

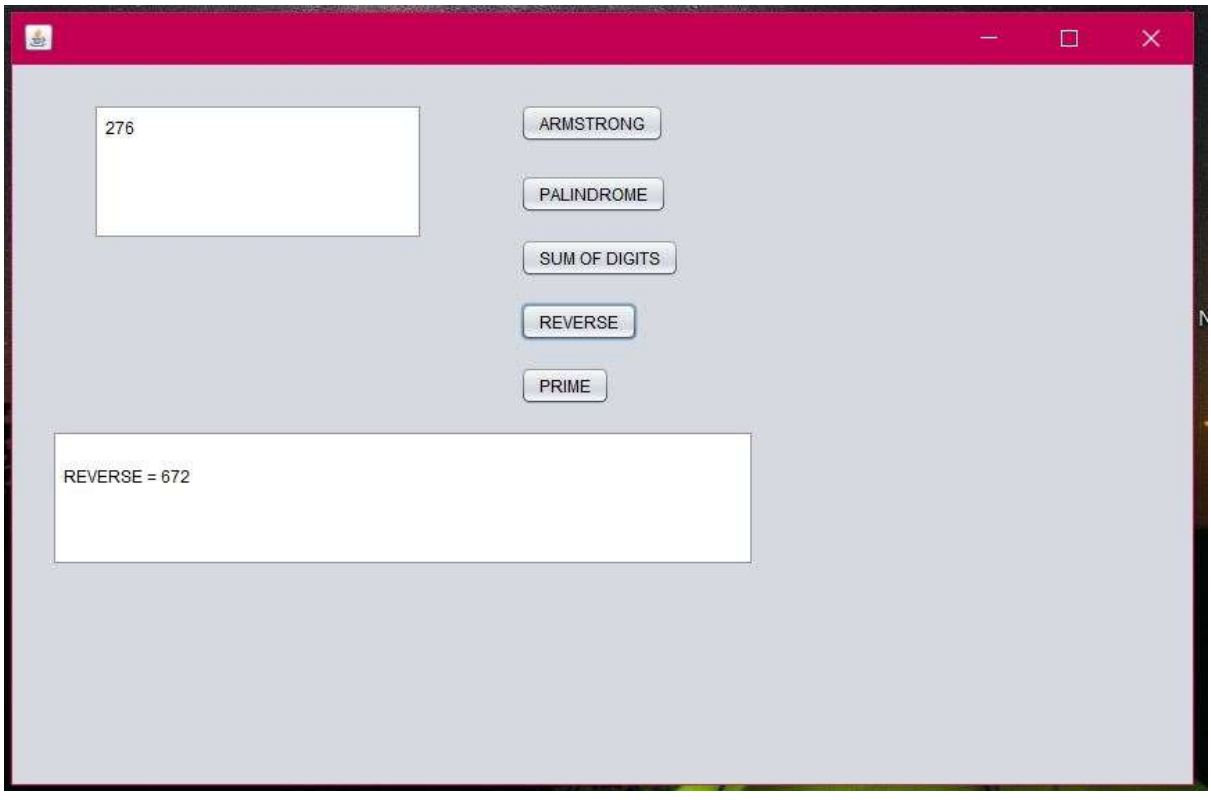
OUTPUT – (CONTINUE)



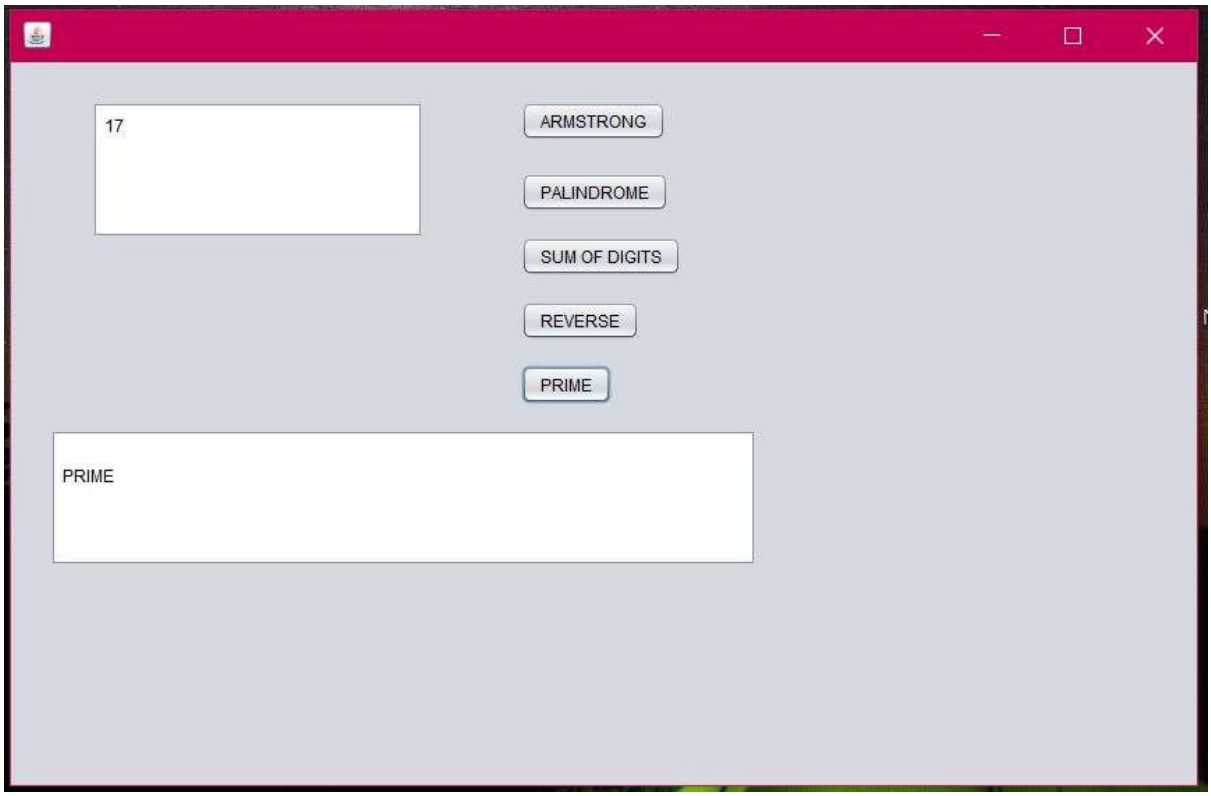


OUTPUT – (CONTINUE)





OUTPUT – (CONTINUE)



IV LOGIN SYSTEM IN SWING

ALGORITHM

1. Start the program.
2. Use jlabel, jbutton, jtextfield, jPasswordField in the program.
3. End the program.

SOURCE CODE

```
public class Slogin extends javax.swing.JFrame {
    public Slogin() {
        initComponents();
    }    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        jTextField1 = new javax.swing.JTextField();
        jPasswordField1 = new javax.swing.JPasswordField();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jProgressBar1 = new javax.swing.JProgressBar();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jLabel1.setText("Username");
        jLabel2.setText("Password");
        jButton1.setText("Login");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }        });
        jButton2.setText("Reset");
```

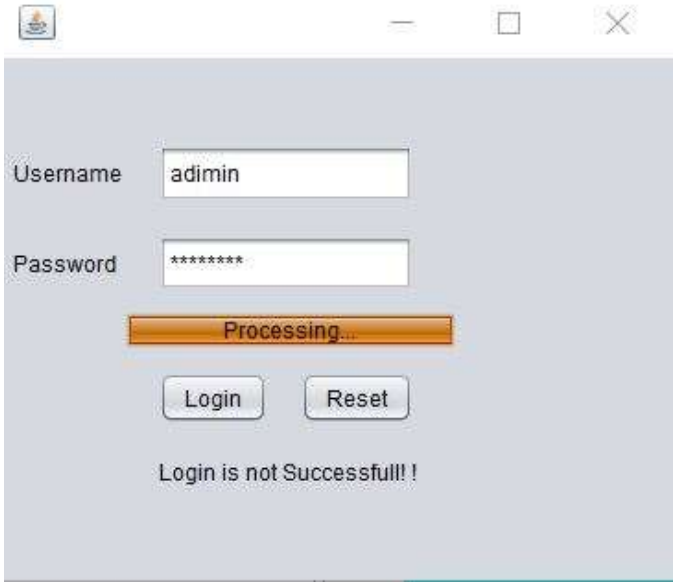
```
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);           }
        });       javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)


.addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)                      .addComponent(jLabel1)
                        .addComponent(jLabel2))
                    .addGap(20, 20, 20)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)                      .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 161,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING, false)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jButton1)
                            .addGap(18, 18, 18)
                            .addComponent(jButton2))
                        .addComponent(jPasswordField1)
                        .addComponent(jTextField1))))
.addComponent(jProgressBar1,
javax.swing.GroupLayout.PREFERRED_SIZE, 179,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

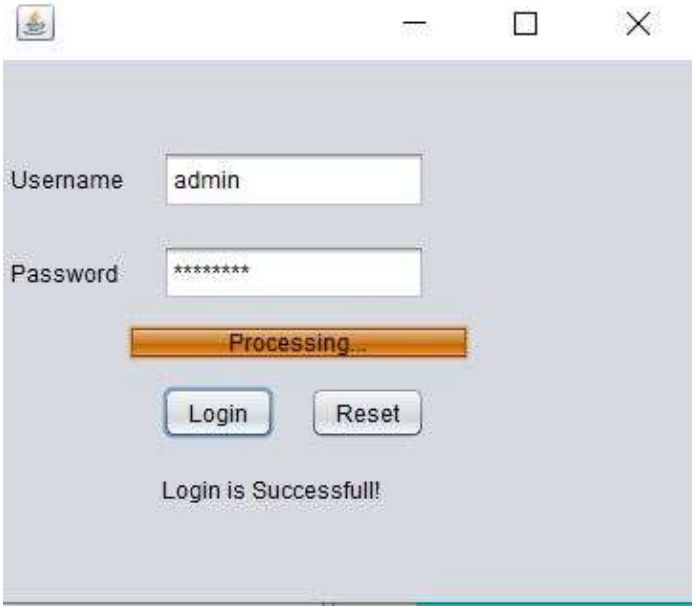
```
            .addContainerGap(118, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(47, 47, 47)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)


  .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
31, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel1))
                .addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jPasswordField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)


 .addComponent(jLabel2))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
  .addComponent(jProgressBar1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(13, 13, 13)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)                    .addComponent(jButton1)
                    .addComponent(jButton2))
                .addGap(18, 18, 18)
                .addComponent(jLabel3)
                .addContainerGap(67, Short.MAX_VALUE))
        );        pack();
```

```
    }// </editor-fold>  private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
jProgressBar1.setString("Processing...");
        jProgressBar1.setStringPainted(true);


 jProgressBar1.setValue(100);
         String uname="admin";String upass="admin123";
         String name=jTextField1.getText();
        String pass=jPasswordField1.getText();
        if(name.equals(uname)&&pass.equals(upass))
            jLabel3.setText("Login is Successfull!");
        else
            jLabel3.setText("Login is not Successfull! !");
    }          private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
 jProgressBar1.setValue(0);
         jPasswordField1.setText("");
         jTextField1.setText("");
jLabel3.setText("");
    }    try {


 for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;              }           }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Slogin.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(Slogin.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Slogin.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Slogin.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } //</editor-fold>/* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Slogin().setVisible(true);
            }});    }    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPasswordField jPasswordField1;
    private javax.swing.JProgressBar jProgressBar1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

OUTPUT





V GPA CALCULATOR

ALGORITHM

1. Start the program.
2. Use jlabel, jbutton, jtextfield, jPasswordField in the program.
3. End the program.

SOURCE CODE

```
 import javax.swing.JOptionPane;
public class Question3 extends javax.swing.JFrame
{
        public Question3() {
        initComponents();
    }
     @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
       jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jTextField3 = new javax.swing.JTextField();
        jTextField5 = new javax.swing.JTextField();
        jTextField6 = new javax.swing.JTextField();
        jTextField4 = new javax.swing.JTextField();
        jTextField7 = new javax.swing.JTextField();
        jTextField8 = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jTextField01 = new javax.swing.JTextField();
```

```
        jTextField02 = new javax.swing.JTextField();
        jTextField03 = new javax.swing.JTextField();
        jTextField04 = new javax.swing.JTextField();
        jTextField06 = new javax.swing.JTextField();
        jTextField05 = new javax.swing.JTextField();
        jTextField08 = new javax.swing.JTextField();
        jTextField07 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jTextField9 = new javax.swing.JTextField();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jTextField2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField2ActionPerformed(evt);
            }
        });

        jLabel1.setText("SUBJECT 2");

        jLabel2.setText("SUBJECT 1");

        jLabel3.setText("SUBJECT 4");

        jLabel4.setText("SUBJECT 3");

        jLabel5.setText("SUBJECT 6");

        jLabel6.setText("SUBJECT 5");

        jLabel7.setText("LAB 1");
```

```
        jLabel8.setText("LAB 2 ");

        jLabel9.setText("GRADE");

        jLabel10.setText("CREDIT");


        jTextField02.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField02ActionPerformed(evt);
            }
        });


        jButton1.setText("Calculate !");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });


        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)
                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
```

```
                .addGap(126, 126, 126)
                .addComponent(jLabel9)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
70, Short.MAX_VALUE)
                .addComponent(jLabel10)
                .addGap(25, 25, 25))
            .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
                .addGap(24, 24, 24)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jLabel3)
                    .addComponent(jLabel1)
                    .addComponent(jLabel4)
                    .addComponent(jLabel5)
                    .addComponent(jLabel6)
                    .addComponent(jLabel2)
                    .addComponent(jLabel7)
                    .addComponent(jLabel8))
                .addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING, false)
                    .addComponent(jTextField1)
                    .addComponent(jTextField2)
                    .addComponent(jTextField3)
                    .addComponent(jTextField4)
                    .addComponent(jTextField5)
                    .addComponent(jTextField6)
```

```
                    .addComponent(jTextField8)
                    .addComponent(jTextField7,
javax.swing.GroupLayout.PREFERRED_SIZE, 77,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jTextField02)
                    .addComponent(jTextField01)
                    .addComponent(jTextField03)
                    .addComponent(jTextField04)
                    .addComponent(jTextField06)
                    .addComponent(jTextField05,
javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jTextField08,
javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jTextField07,
javax.swing.GroupLayout.Alignment.TRAILING))))
            .addContainerGap())
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addComponent(jButton1)
                    .addGap(71, 71, 71))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
```

```
                    .addComponent(jTextField9,
javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(88, 88, 88))))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(22, 22, 22)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel9)
                    .addComponent(jLabel10))
                .addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2)
                    .addComponent(jTextField01,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabel1)

                .addComponent(jTextField02,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)


.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)

                .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabel4)

                .addComponent(jTextField03,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)


.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)

                .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabel3)

                .addComponent(jTextField04,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel6)
                .addComponent(jTextField05,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                .addComponent(jTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel5)
                .addComponent(jTextField06,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                .addComponent(jLabel7)
                .addComponent(jTextField7,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jTextField07,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel8)
                    .addComponent(jTextField8,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField08,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
                .addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
31, Short.MAX_VALUE)
                .addComponent(jTextField9,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(19, 19, 19))
        );


        pack();
    }// </editor-fold>


    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
```

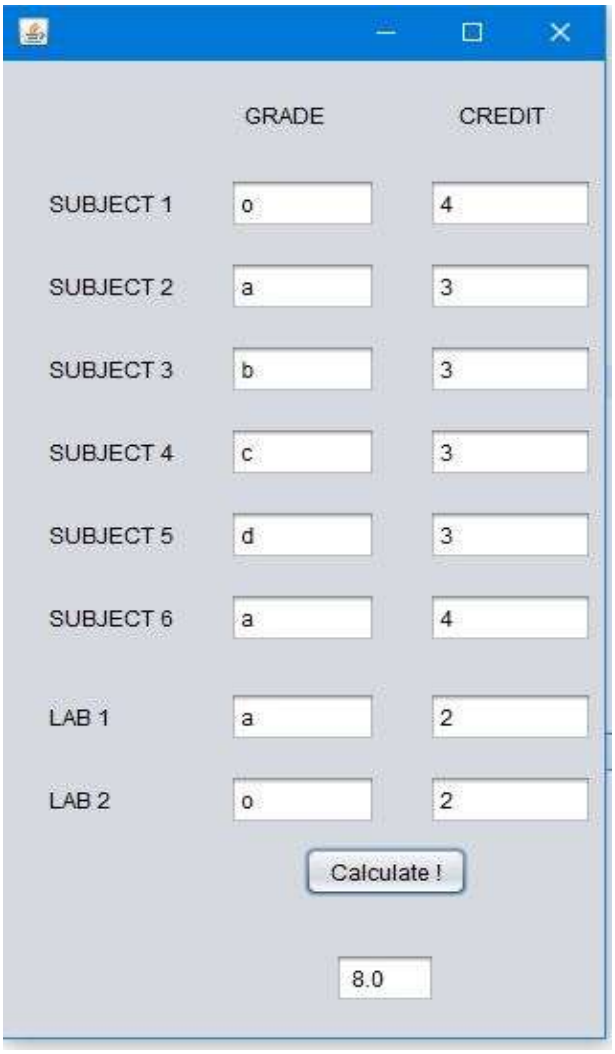
```
}

private void jTextField02ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {


            int csum=0;
            int val=1;
            float GPA;
            int a[]=new int[9];
            int c[]=new int[9];
            char[] m= new char[8];
        a[0]=Integer.parseInt(jTextField01.getText());
        a[1]=Integer.parseInt(jTextField02.getText());
        a[2]=Integer.parseInt(jTextField03.getText());
        a[3]=Integer.parseInt(jTextField04.getText());
        a[4]=Integer.parseInt(jTextField05.getText());
        a[5]=Integer.parseInt(jTextField06.getText());
        a[6]=Integer.parseInt(jTextField07.getText());
        a[7]=Integer.parseInt(jTextField08.getText());
        for(int i=0;i<8;i++)
            csum=csum+a[i];

        String i1=jTextField1.getText(); m[0]=i1.charAt(0);
        String i2=jTextField2.getText();m[1]=i2.charAt(0);
        String i3=jTextField3.getText();m[2]=i3.charAt(0);
```

```
        String i4=jTextField4.getText();m[3]=i4.charAt(0);
        String i5=jTextField5.getText();m[4]=i5.charAt(0);
        String i6=jTextField6.getText();m[5]=i6.charAt(0);
        String i7=jTextField7.getText();m[6]=i7.charAt(0);
        String i8=jTextField8.getText();m[7]=i8.charAt(0);

        for(int i=0;i<8;i++)
        {
          if(m[i]=='o')
          c[i]=10;
          if(m[i]=='a')
              c[i]=9;
          if(m[i]=='b')
              c[i]=8;
          if(m[i]=='c')
              c[i]=7;
          if(m[i]=='d')
              c[i]=6;
          if(m[i]=='f')
              c[i]=0;
          }
        for(int i=0;i<8;i++)
              val = c[i]*a[i]+val;


        GPA= val /csum;

        jTextField9.setText(" "+GPA);
    }
```

```
 public static void main(String args[]) {
          java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
           new Question3().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JTextField jTextField01;
private javax.swing.JTextField jTextField02;
private javax.swing.JTextField jTextField03;
private javax.swing.JTextField jTextField04;
private javax.swing.JTextField jTextField05;
private javax.swing.JTextField jTextField06;
private javax.swing.JTextField jTextField07;
private javax.swing.JTextField jTextField08;
```

```
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
    // End of variables declaration
}
```

OUTPUT



VI CHANGING IMAGE USING SLIDER

ALGORITHM

1. Start the program.
2. Use jlabel, jbutton, jtextfield, jPasswordField in the program.
3. End the program.

SOURCE CODE

```
    import javax.swing.ImageIcon;
 public class Sslider extends javax.swing.JFrame
{
   String path1="C:\\Users\\student\\OneDrive\\Pictures\\1.png";
   String path2="C:\\Users\\student\\OneDrive\\Pictures\\2.png";
   String path3="C:\\Users\\student\\OneDrive\\Pictures\\3.png";
   ImageIcon img1=new ImageIcon(path1);
   ImageIcon img2=new ImageIcon(path2);
   ImageIcon img3=new ImageIcon(path3);
    public Sslider() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {


        jSlider1 = new javax.swing.JSlider();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


        jLabel1.setText("USE THE SLIDER TO CHANGE THE IMAGE");
```

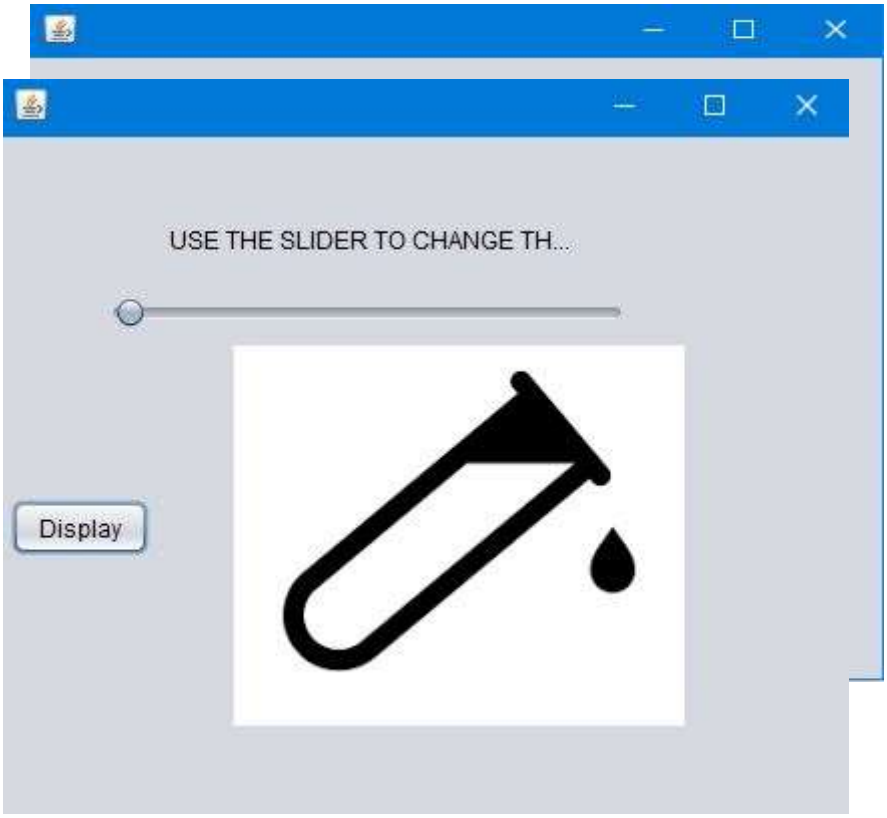
```
        jButton1.setText("Display");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jButton1)
                .addGap(42, 42, 42)
                .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(49, 49, 49))
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(57, 57, 57)
                        .addComponent(jSlider1,
javax.swing.GroupLayout.PREFERRED_SIZE, 253,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
                .addGroup(layout.createSequentialGroup()
                    .addGap(85, 85, 85)
                    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(116, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(43, 43, 43)
                .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
18, Short.MAX_VALUE)
                .addComponent(jSlider1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 190,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addGap(77, 77, 77)
```

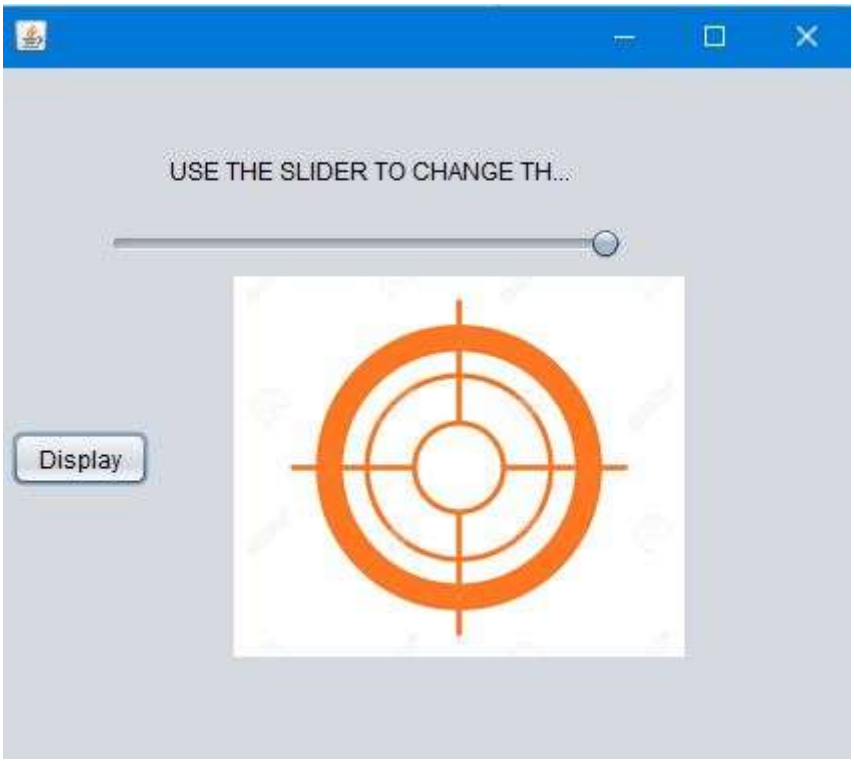
```
                .addComponent(jButton1)
                .addGap(136, 136, 136))))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

       int i=jSlider1.getValue();
       System.out.println(""+i);
       jLabel2.setVisible(true);
        if(i<=30)
        jLabel2.setIcon(img1);
        if((i<=70)&&(i>=31))
        jLabel2.setIcon(img2);
        if((i<=100)&&(i>=71))
        jLabel2.setIcon(img3);

    }



    public static void main(String args[]) {
            java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Sslider().setVisible(true);
            }
        });
    }
```

```
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JSlider jSlider1;
    // End of variables declaration
}
```

OUTPUT



OUTPUT – (CONTINUE)



VII CHARACTER FREQUENCY

ALGORITHM

1. Start the program.
2. Use jlabel, jbutton, jtextfield,jPasswordField in the program.
3. End the program.

SOURCE CODE:

```
package tp;
import javax.swing.JOptionPane;
public class ex3 extends javax.swing.JFrame {
    public ex3() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
```

```
    private void initComponents() {
        jCheckBoxMenuItem1 = new javax.swing.JCheckBoxMenuItem();
        tf1 = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        ta = new javax.swing.JTextArea();
        jLabel2 = new javax.swing.JLabel();
        jCheckBoxMenuItem1.setSelected(true);
        jCheckBoxMenuItem1.setText("jCheckBoxMenuItem1");
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
tf1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                tf1ActionPerformed(evt);
            }
        });        jLabel1.setText("Enter a String:"); jButton1.setText("Check
Frequency");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });   ta.setColumns(20);
        ta.setRows(5);
        jScrollPane1.setViewportView(ta);
        jLabel2.setText("Frequency:");
        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(204, 204, 204)
                    .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 126,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(185, 185, 185)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                        .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addContainerGap(130, Short.MAX_VALUE)
            .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tf1, javax.swing.GroupLayout.PREFERRED_SIZE,
158, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(236, 236, 236))
    );      layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
```

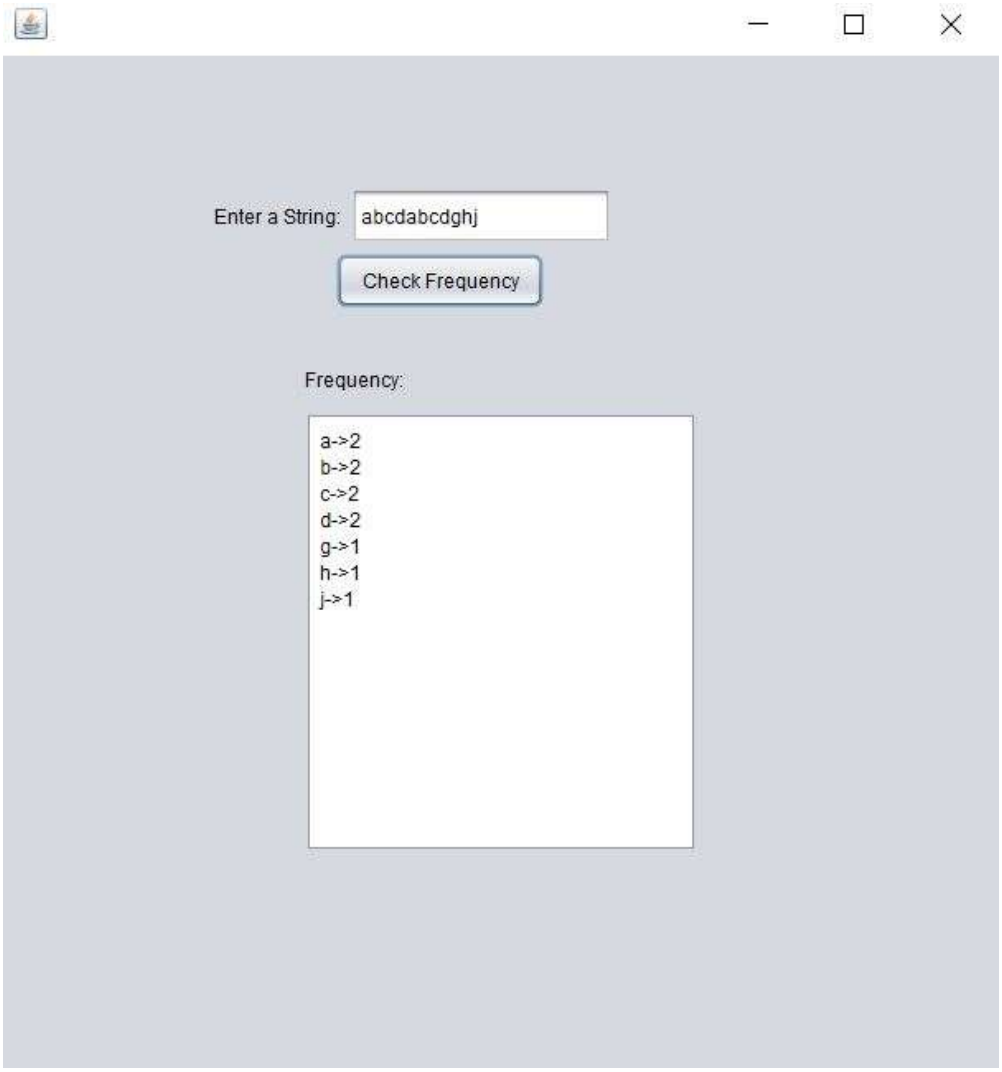
```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(80, 80, 80)
                    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(tf1,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(30, 30, 30)
            .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 267,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(134, Short.MAX_VALUE))
        );        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void tf1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_tf1ActionPerformed
        // TODO add your handling code here:
```

```
    }//GEN-LAST:event_tf1ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
        String s = tf1.getText();String f,w,res;
        int l = s.length();
        char c[];
        c=s.toCharArray();
        int i;
        int freq[] = new int[l];
        for(i=0;i<l;i++)
        {           freq[i]=1;
          for(int j=i+1;j<l;j++)
          {
             if(c[i]==c[j]){
                freq[i]++;
                c[j]='0';
          }
        }}
        for(i=0;i<freq.length;i++)
        {           if(c[i]!=' ' && c[i]!='0')
           {            f=Integer.toString(freq[i]);
            w=Character.toString(c[i]);
            res=w+"->"+f+"\n";
            ta.append(res);}}
        }
    }//GEN-LAST:event_jButton1ActionPerformed
    public static void main(String args[]) {
            try {
```

```
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }       } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ex3.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ex3.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ex3.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ex3.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        }       //</editor-fold>
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ex3().setVisible(true);
            }
        });
    }    //Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JCheckBoxMenuItem jCheckBoxMenuItem1;
    private javax.swing.JLabel jLabel1;
```

```
    private javax.swing.JLabel jLabel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea ta;
    private javax.swing.JTextField tf1;
    // End of variables declaration//GEN-END:variables}
```

OUTPUT



VIII DROPDOWN LISTBOX

ALGORITHM

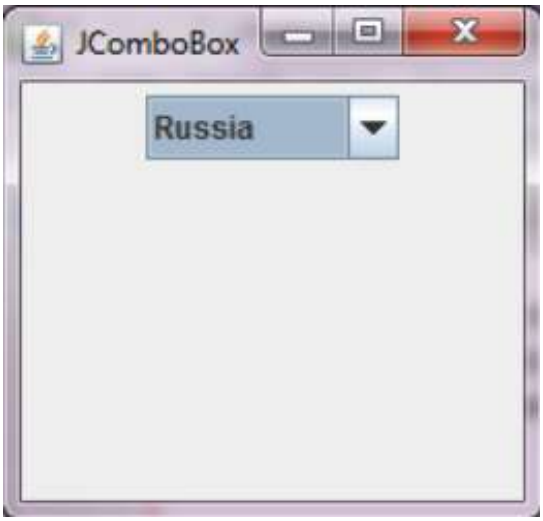
1. Start the program.
2. Use jcombobox, jlabel in the program.
3. End the program.

SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class Combo4
{
public static void main(String... ar)
{
SwingUtilities.invokeLater(new Runnable() {
public void run()
{
new A();
}
});

}//Closing the main method
}//Closing the class Combo4


class A implements ActionListener
```

```
{
String [] BRICS;
JFrame jf;
JComboBox<String> combo;
JLabel label1;

A()
{
BRICS = new String[]{"Russia", "India", "South Africa", "Brazil", "China"};
if= new JFrame("JComboBox");
combo= new JComboBox<String>(BRICS);
label1 = new JLabel();

jf.add(combo);
combo.addActionListener(this);

jf.setLayout(new FlowLayout());
jf.setSize(210,200);
jf.setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
JComboBox cb = (JComboBox)ae.getSource();
label1.setText( ((String)cb.getSelectedItem()) + " is selected");
jf.add(label1);
jf.setVisible(true);}}
```

OUTPUT



RESULT

Thus swing GUI applications are implemented using java programs.

EX.NO: 9 **JAVA THREADING**

AIM

To implement the threads in java.

I SINGLE THREAD

ALGORITHM

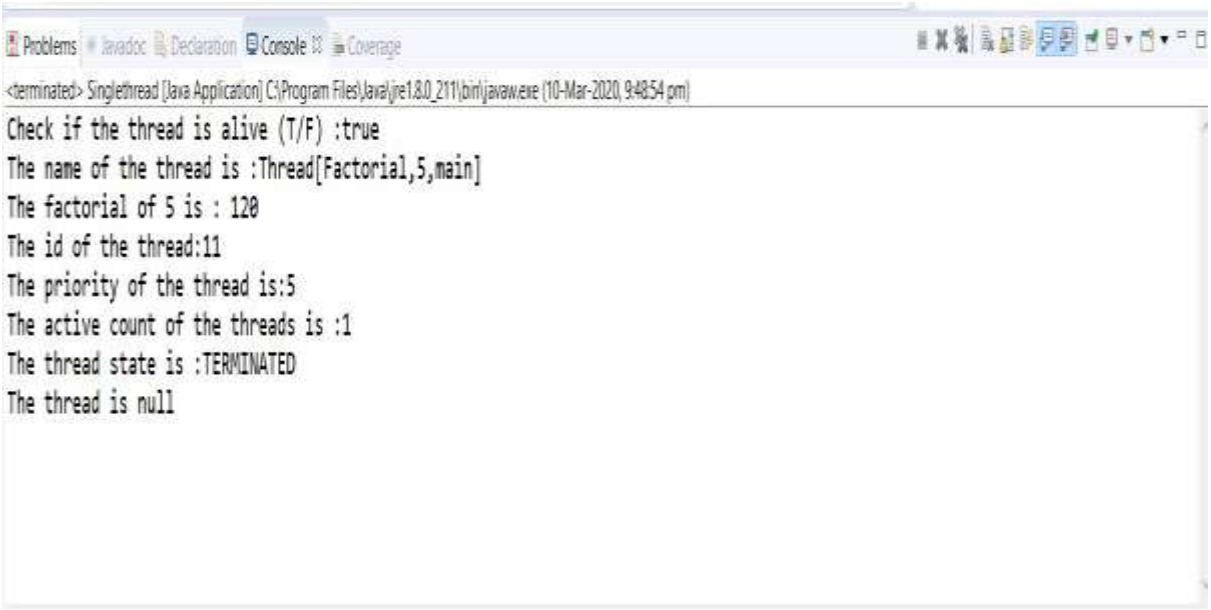
1. Start the program.
2. Extend the thread class.
3. End the program.

SOURCE CODE

```
package class_and_objects;

public class Singlethread extends Thread
{

        public void run() {
                int i=1,f=1;
                for(i=0;i<5;i++) {
                        f=f*(i+1);
                }
                System.out.println("The factorial of 5 is : "+f);
              }

              public static void main(String[] args) {

                    Singlethread t=new Singlethread();
                    t.start();
                  System.out.println("Check if the thread is alive (T/F) :"+
t.isAlive());
                  t.setName("Factorial");
                  System.out.println("The name of the thread is :"+t);

                  System.out.println("The id of the thread:"+t.getId());
                  System.out.println("The priority of the thread
is:"+t.getPriority());
                  System.out.println("The active count of the threads is
:"+t.activeCount());
                  System.out.println("The thread state is :"+t.getState());
                  System.out.println("The thread is "+t.getThreadGroup());

                    }
```

```
}
```

OUTPUT



II ASYNCHRONOUS MULTITHREADING

ALGORITHM

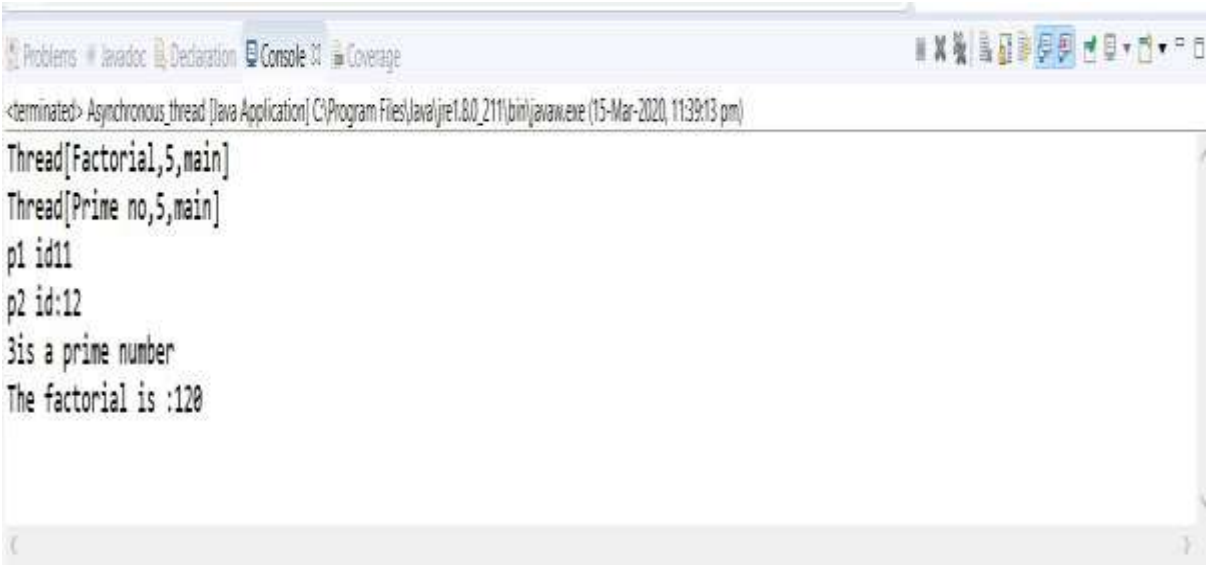
1. Start the program.
2. Implement multithreading in asynchronous way.
3. End the program.

SOURCE CODE:

```
package class_and_objects;

class  Person1 extends Thread{
        public void run() {
                try {
                        Thread.sleep(100);
                } catch (InterruptedException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                int i=0;int f=1;
                for(i=0;i<5;i++){
                        f=f*(i+1);
                }
                        System.out.println("The factorial is :"+f);
```

```
	}
}
class  Person2  extends Thread{
	public void run() {

		int n=3,m=0,flag=0,i;
		m=n/2;
		for(i=2;i<=m;i++) {
			if(n%i==0) {
				System.out.println(n+"is not a prime no");
				flag=1;
				break;
			}
		}
		if(flag==0) {
			System.out.println(n+"is a prime number");
		}

	}
}
public class  Asynchronous_thread {
	static void Tname() {
		Thread t=Thread.currentThread();
		String name=t.getName();
		System.out.println("Thread name:\t"+name);
		}

	public static void main(String[] args) throws InterruptedException {
		Person1 p1=new Person1();
		Person2 p2=new Person2();
		p1.setName("Factorial");
		p2.setName("Prime no");
	System.out.println(p1);
	System.out.println(p2);
		System.out.println("p1 id"+p1.getId());
		System.out.println("p2 id:"+p2.getId());
		p1.start();
		p2.start();



	}

}
```

OUTPUT



III SYNCHRONOUS MULTITHREADING

ALGORITHM

1. Start the program.
2. Implement synchronized method of multithreading in the program.
3. End the program.

SOURCE CODE

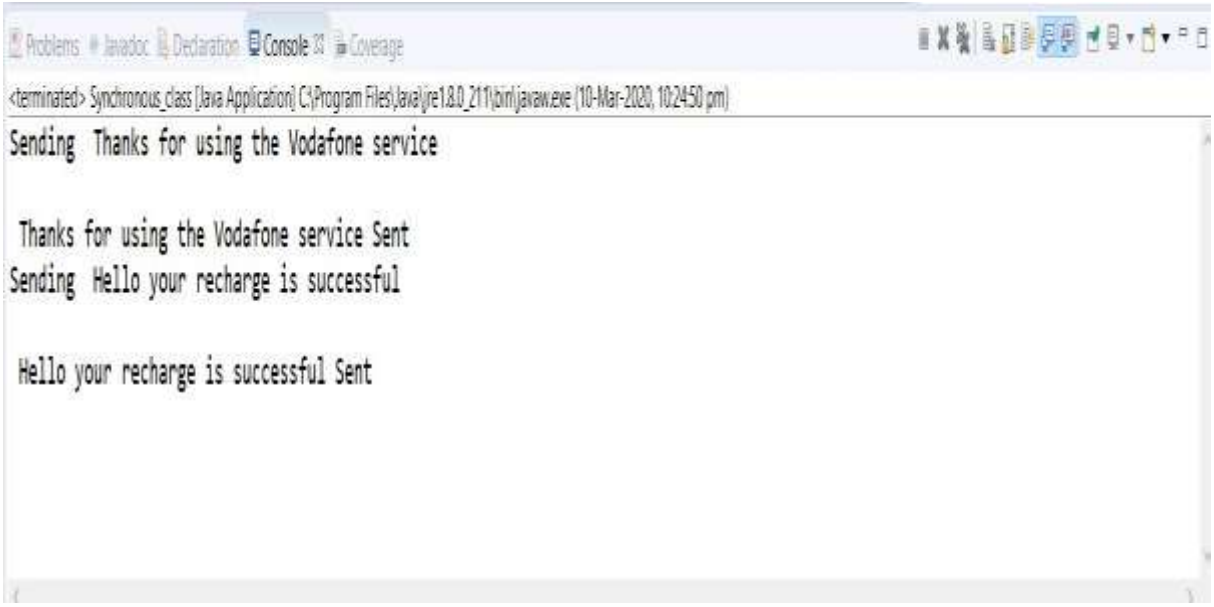
```
package class_and_objects;


import java.io.*;
import java.util.*;


// A Class used to send a message
class Sender
{
    public synchronized void send(String msg)
    {
        System.out.println("Sending\t"  + msg );
```

```
        try
        {
            Thread.sleep(1000);
        }
        catch (Exception e)
        {
            System.out.println("Thread  interrupted.");
        }
        System.out.println("\n" + msg + "Sent");
    }
}

// Class for send a message using Threads
class ThreadedSend extends Thread
{
    private String msg;
    Sender  sender;

    // Recieves a message object and a string
    // message to be sent
    ThreadedSend(String m,  Sender obj)
    {
        msg = m;
        sender = obj;
    }

    public void run()
    {
        // Only one thread can send a message
```

```
        // at a time.
        synchronized(sender)
        {
            // synchronizing the snd object
            sender.send(msg);
        }
    }
}

// Driver class
public class Synchronous_class
{
    public static void main(String args[])
    {
        Sender snd = new Sender();
        ThreadedSend S1 =
            new ThreadedSend( " Thanks for using the Vodafone service "  , snd );
        ThreadedSend S2 =
            new ThreadedSend(" Hello your recharge is successful " , snd );

        // Start two threads of ThreadedSend type
        S1.start();
        S2.start();

        // wait for threads to end
        try
        {
            S1.join();
            S2.join();
```

```
        }
        catch(Exception e)
        {
            System.out.println("Interrupted");
        }
    }
}
```

OUTPUT



IV ASYNCHRONOUS MULTITHREADING

USING JOIN

ALGORITHM

1. Start the program.
2. Implement multithreading in asynchronous way using join method.
3. End the program.

SOURCE CODE

```
package class_and_objects;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
public class ThreadJoinExample
{
	public static void main(String[] args)
	{
		Integer[] values = new Integer[] { 3, 1, 14, 3, 4, 5, 6, 7, 8, 9, 11,
				3, 2, 1 };
		Average avg = new Average(values);
		Median median = new Median(values);
		Thread t1 = new Thread(avg, "t1");
		Thread t2 = new Thread(median, "t2");
		System.out.println("Start the thread t1 to calculate average");
		t1.start();
		System.out.println("Start the thread t2 to calculate median");
		t2.start();
		try {
			System.out.println("Join on t1");
			t1.join();
			System.out.println("t1 has done with its job of calculating
			average");
		} catch (InterruptedException e) {
			System.out.println(t1.getName() + " interrupted");
		}
		try {
			System.out.println("Join on t2");
```

```
                t2.join();
                System.out
                        .println("t2 has done with its job of calculating
median");
            } catch (InterruptedException e) {
                System.out.println(t2.getName() + " interrupted");
            }
            System.out.println("Average: " + avg.getMean() + ", Median: "
                    + median.getMedian());
        }

        /**
         * Calculate average of numbers. Sum all the int values and divide it by
         * total count.
         */
        private static class Average implements Runnable {
            private Integer[] values;
            private int mean;

            Average(Integer[] values) {
                this.values = values;
            }

            @Override
            public void run() {
                mean = 0;
                int n = values.length;
                for (int i : values) {
                    mean += i;
```

```
                }
                mean /= n;
        }

        public int getMean() {
                return mean;
        }
}

/**
 * Sorts the given int list and calculates the median value. If size is
 * even, the mean of middle and middle-1.
 *
 */
private static class Median implements Runnable {
        private Integer[] values;
        private int median;

        Median(Integer[] values) {
                this.values = values;
        }

        @Override
        public void run() {
                List<Integer> sortedList = Arrays.asList(values);
                Collections.sort(sortedList);
                int n = values.length;
                int middle = n / 2;
                if (n % 2 == 0) {
```

```
                        median = (sortedList.get(middle) +
sortedList.get(middle - 1)) / 2;
                } else {
                        median = sortedList.get(middle);
                }
        }

        public int getMedian() {
                return median;
        }

}
```

OUTPUT

V ASYNCHRONOUS MULTITHREADING WITH PRIORITY

ALGORITHM

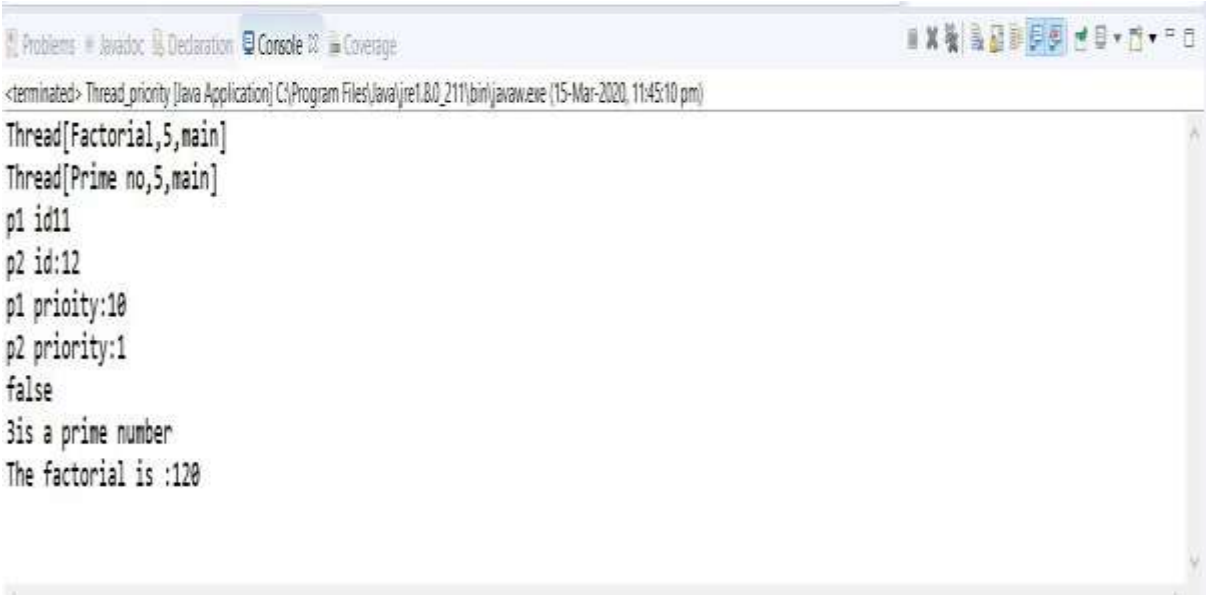
1. Start the program.
2. Implement multithreading in asynchronous way with priority.
3. End the program.

SOURCE CODE

```
package class_and_objects;

 class P1 extends Thread {

		public void run() {
			try {
				Thread.sleep(100);
			} catch (InterruptedException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
			int i=0;int f=1;
			for(i=0;i<5;i++){
				f=f*(i+1);
			}
				System.out.println("The factorial is :"+f);
		}
	}
	class  P2  extends Thread{
		public void run() {

			int n=3,m=0,flag=0,i;
			m=n/2;
			for(i=2;i<=m;i++) {
				if(n%i==0) {
					System.out.println(n+"is not a prime no");
					flag=1;
					break;
				}
			}
			if(flag==0) {
				System.out.println(n+"is a prime number");
			}

		}
	}
```

```
public class Thread_priority
{
        static void Tname() {
                Thread t=Thread.currentThread();
                String name=t.getName();
                System.out.println("Thread name:\t"+name);
                }

        public static void main(String[] args) throws
InterruptedException {
                P1 p1=new P1();
                P2 p2=new P2();
                p1.setName("Factorial");
                p2.setName("Prime no");
        System.out.println(p1);
        System.out.println(p2);
                System.out.println("p1 id"+p1.getId());
                System.out.println("p2 id:"+p2.getId());
                p1.start();
                p2.start();
                p1.setPriority(Thread.MAX_PRIORITY);
                p2.setPriority(Thread.MIN_PRIORITY);
                System.out.println("p1 prioity:"+p1.getPriority());
                System.out.println("p2 priority:"+p2.getPriority());
                System.out.println(p2.isInterrupted());


        }

}
```

OUTPUT



RESULT

Thus threading is implemented using java.

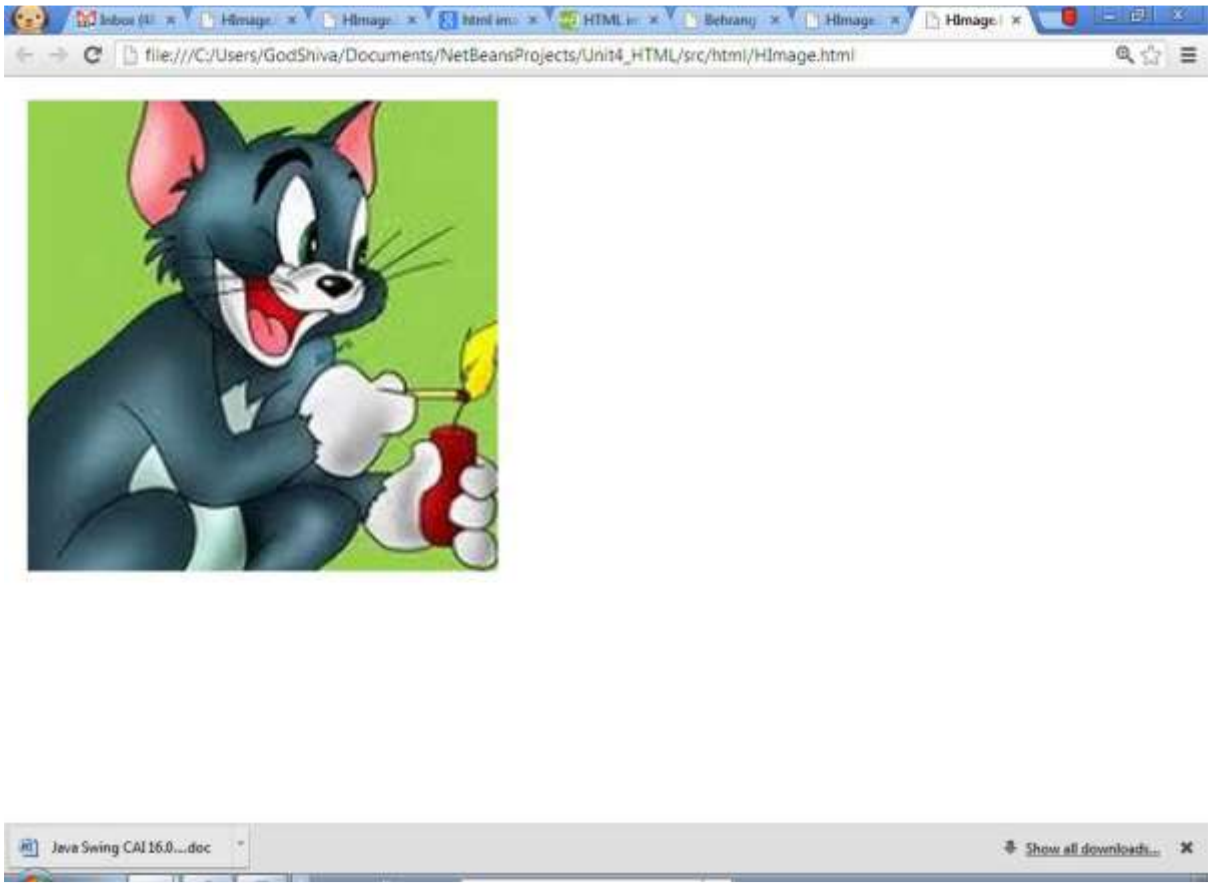
EX.NO: 10 SIMPLE WEBSITES USING HTML5 TAGS

I. EXAMPLE OF IMAGE DEMO

SOURCE CODE

```
<html>
<body>
<img src=" C:\Users\GodShiva\Desktop\pussycat.jpg"
alt="pussy cat" align="left">
</body>
</html>
```

OUTPUT



II. EXAMPLE OF HYPER LINKS

(Hmain.html)

1. SOURCE CODE

```
<html>
<body>
To see the wall of cricket:
<a href="Dravid.html">
Click Here
</a>
</body>
</html>
```

(Dravid.html)

```
<html>
<body>
<center>
<img src="C:\Users\GodShiva\Desktop\TestHero.jpg"
alt="Ragul Dravid" width="500" height="500">
</center>
</body>
</html>
```

OUTPUT- HOME PAGE

DISPLAYING PAGE VIA LINKS

III. EXAMPLE OF HTML TABLE

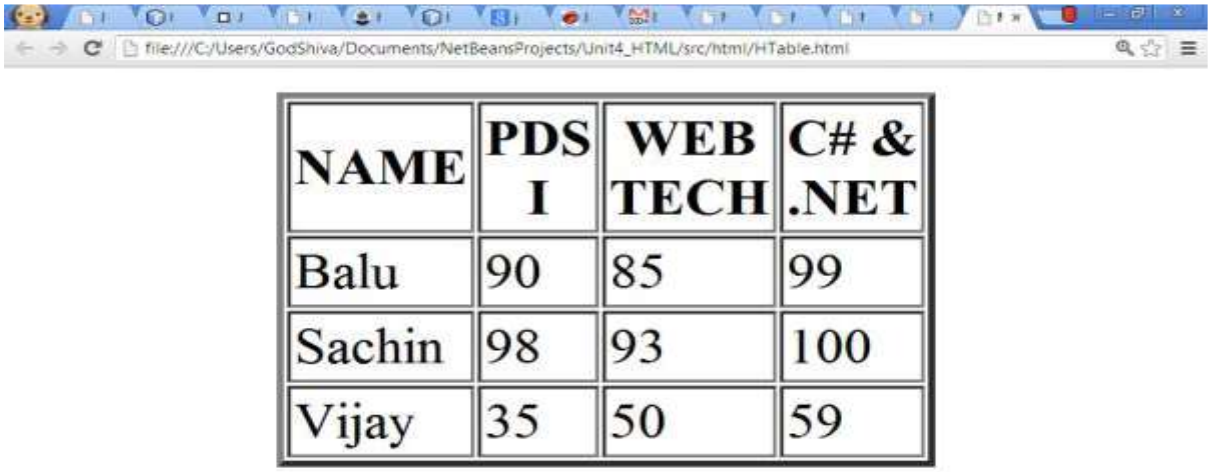
SOURCE CODE

```
<html>
<head>
<title>MARK LIST</title>
</head>
<body>
<center><table WIDTH="100" border="2" CELLSPACING="1">
<tr>
<th>NAME<th>PDS I<TH>WEB TECH<th>C# & .NET
</tr>
<tr>
| HTML SoftNotes 20 March 2019, 12:48:12 PM |
```

26

```
<td>Balu</td><td>90</td><td>85<td>99</td>
</tr>
<tr>
<td>Sachin</td><td>98<td>93<td>100</td>
</tr>
<tr>
<td>Vijay</td><td>35</td><td>50</td><td>59</td>
</tr>
</table></center>
</body>
</html>
```

OUTPUT

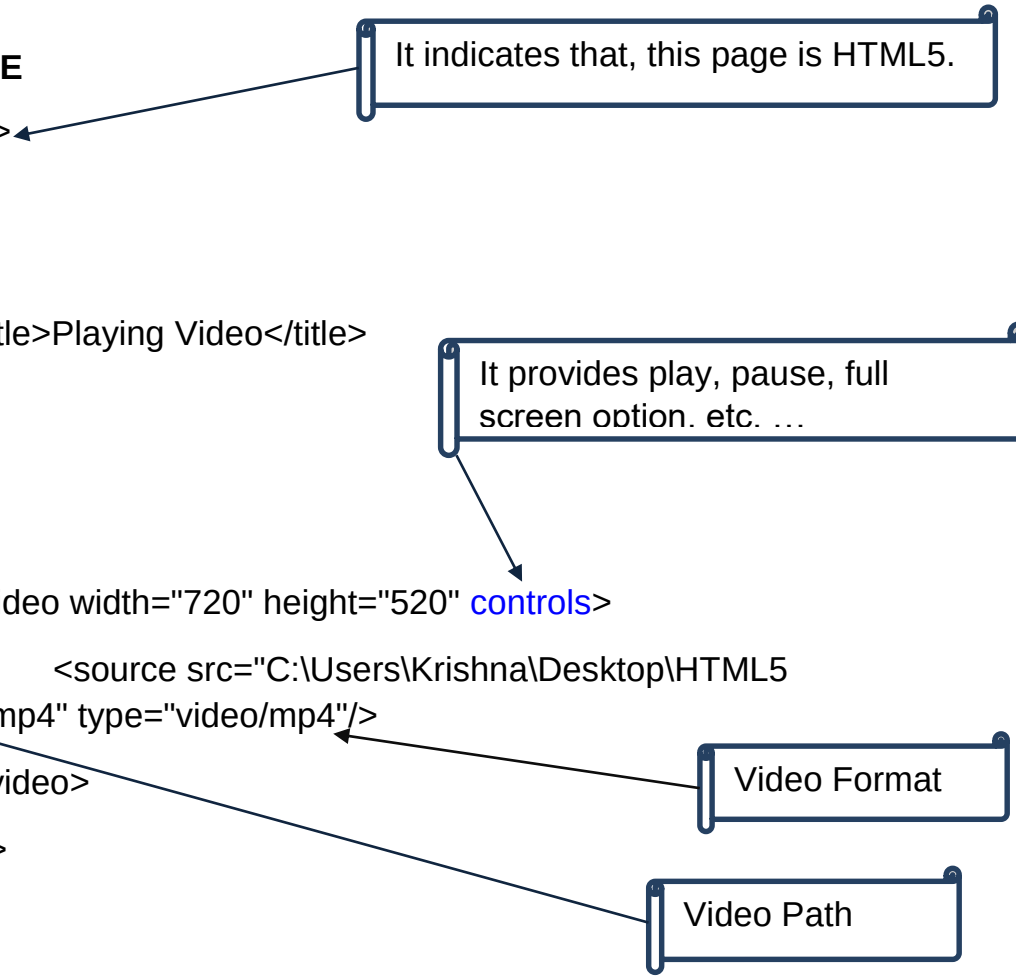


IV. PLAYING A VIDEO USING HTML 5

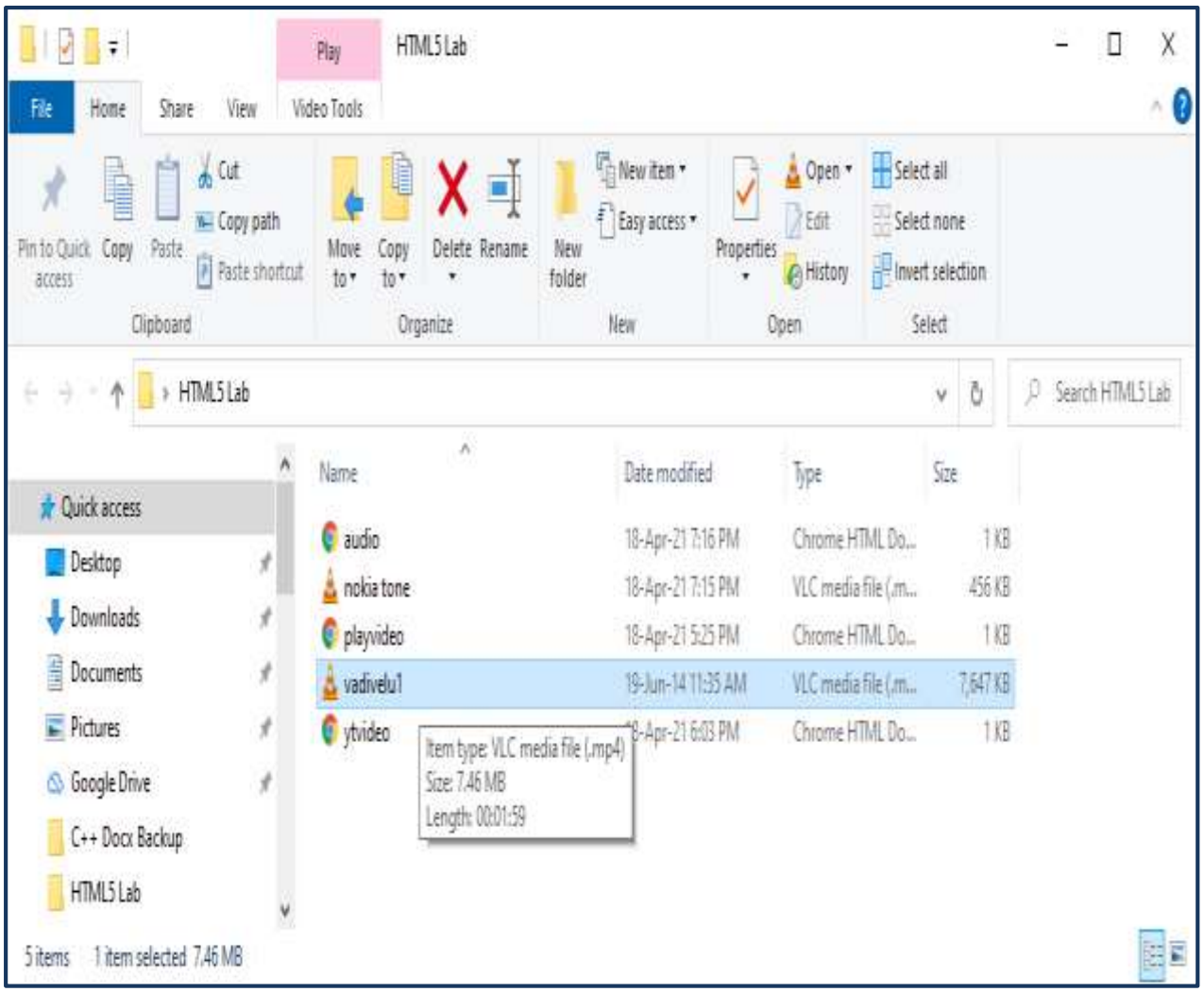
Application Type	:	**Web Application**
Language Used	:	**HTML**
Technology	:	**HTML5**
Editor	:	**Notepad ++**
Platform	:	**Windows 10**

SOURCE CODE

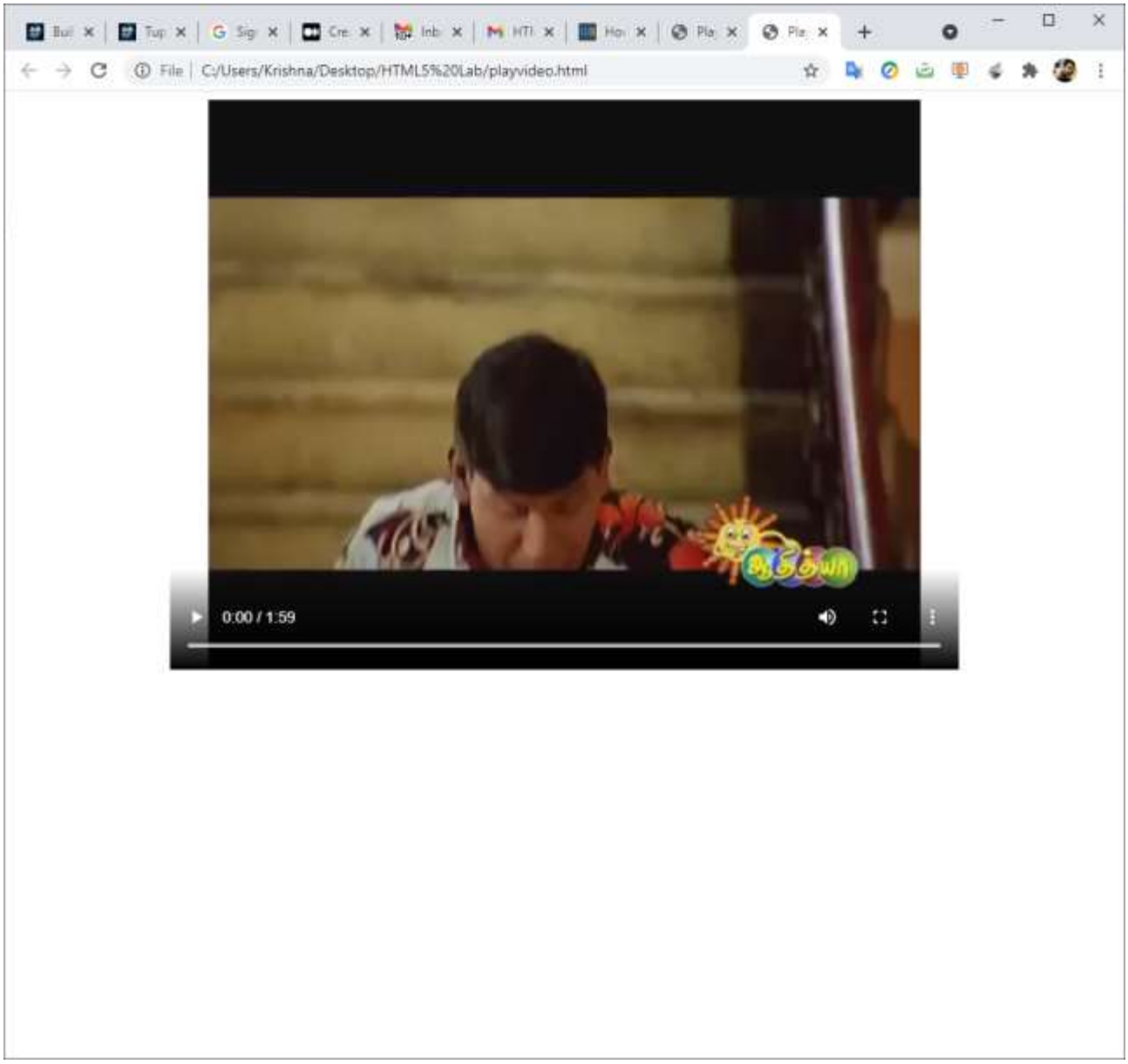
```
<!doctype html>
<html>
    <head>
        <title>Playing Video</title>
    </head>
    <body>
    <center>
        <video width="720" height="520" controls>
            <source src="C:\Users\Krishna\Desktop\HTML5
Lab\vadivelu1.mp4" type="video/mp4"/>
        </video>
    </center>
    </body>
</html>
```



LOCATION OF INPUT VIDEO

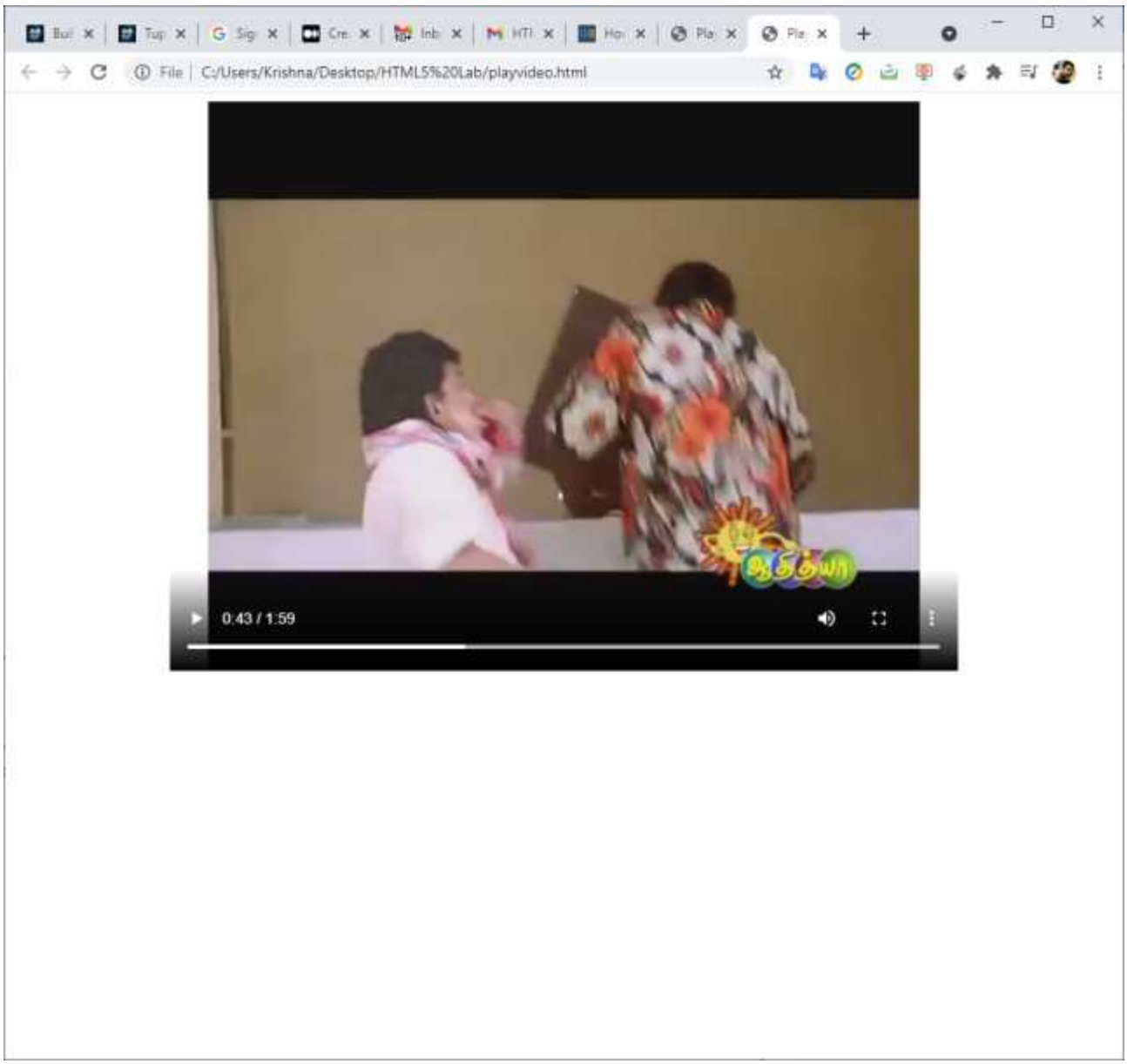


OUTPUT - HOME PAGE



PLAYING VIDEO

PAUSING A VIDEO



RESULT

Thus the websites creation using HTML5 tags are implemented and tested successfully.

EX.NO: 11 **WEB FORM CREATION AND VALIDATION USING JAVA SCRIPTS**

I. FORM SUBMISSION EXAMPLE

OS : Windows 10

Tested IDE : Netbeans 8.2

Application : HTML5 Web Application

(coursereg.html)

SOURCE CODE

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body
      {
        background-color: #00b1c1;
        text-align: center;
      }
      .dd
      {
        padding:2%;
        color: white;
        background-color: #4f5b65;
        width: 96%;
        font-size: 21px;
        border-color: #ef9421;
        border-style:solid;
        align-self: center
      }
      h3
      {
        color: white;
```

Internal CSS

Design the <body> using CSS

Design the tag using CSS

Design the <h3> tag using internal CSS

```
        font-size: 42px;
        margin: 2px;
    }
    .ss
    {
        background-color: #ffff00;
        border: none;
        color: black;
        padding: 9px 24px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
    }
    .rr
    {
        background-color: #00b1c1;
        border: none;
        color: whitesmoke;
        padding: 9px 24px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
    }
</style>
<script type="text/javascript">
    function disp()
    {
        var f1,f2,f3,f4;
// accessing HTML form Input Elements using getElementId() method
```

Design the submit button using internal CSS

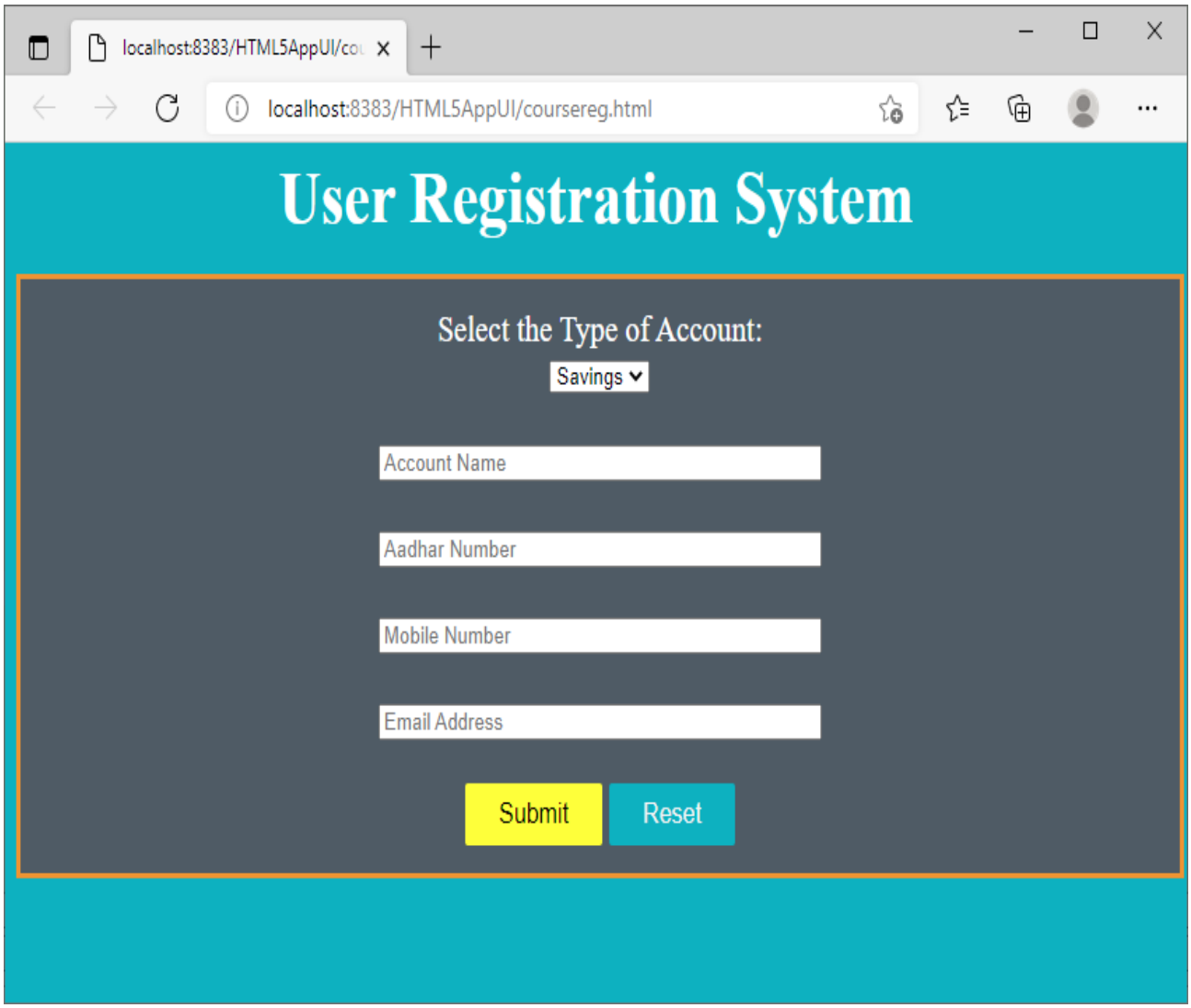
Design the reset button using internal CSS

```
        v1=document.getElementById("ac");
        v2=document.getElementById("b1");
        v3=document.getElementById("b2");
        v4=document.getElementById("b3");
        v5=document.getElementById("b4");
    // get the selected value from dropdown list
        acctype=v1.value;
    // checking whether the name text box is empty or not
        if(v2.value!=="")
        {
           f1=true;
        }
        else
        {
           alert("Name is Empty");
           f1=false;
        }
    // checking whether the aadhar text box is empty or not
        if(v3.value!=="")
        {
           f2=true;
        }
        else
        {
           alert("Aadhar Numeber is Empty");
           f2=false;
        }
    // checking whether the mobile number text box is empty or not
        if(v4.value!=="")
        {
           f3=true;
```

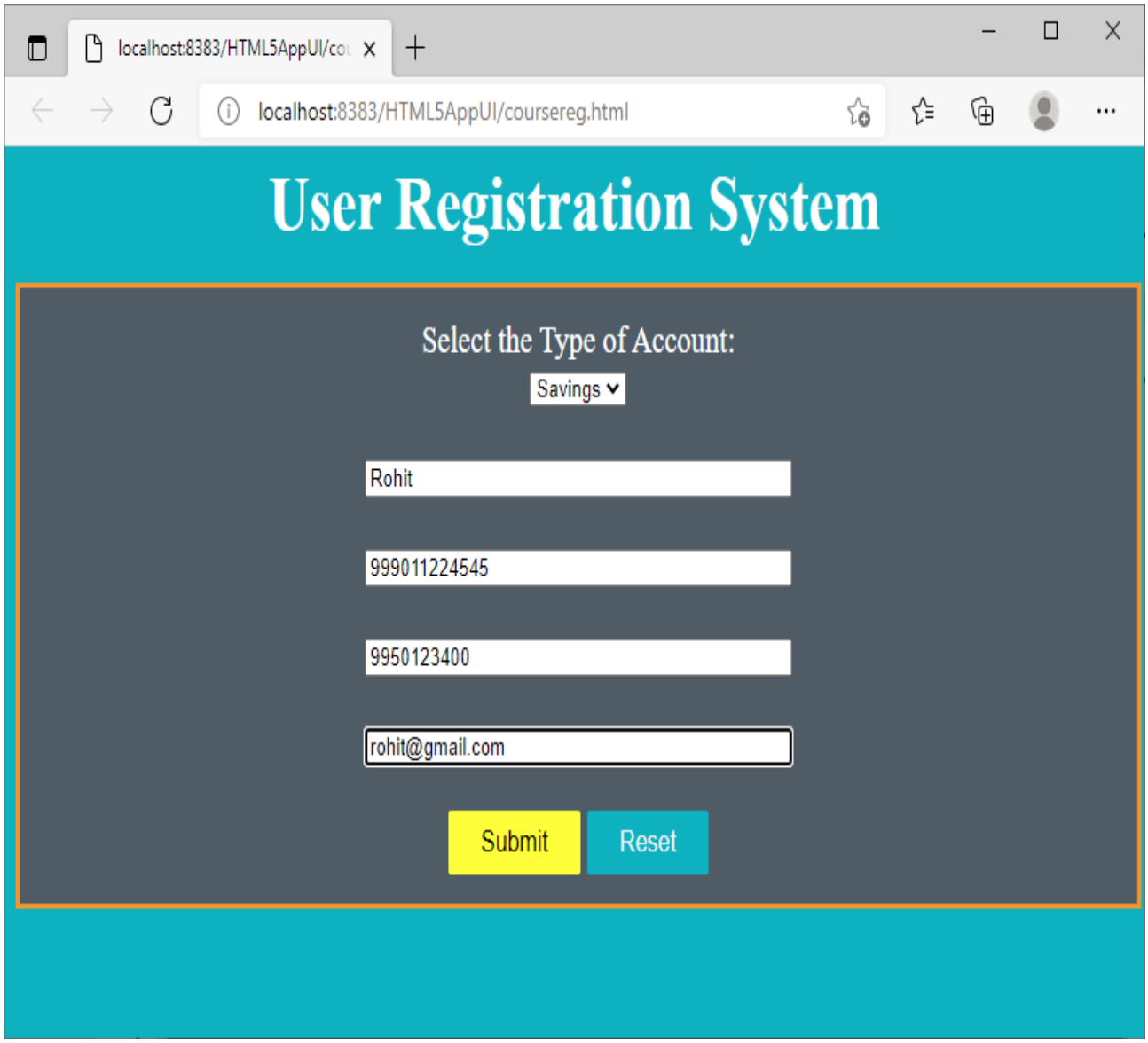
```
            }
            else
            {
                alert("Mobile Number is Empty");
                f3=false;
            }
        // checking whether the email text box is empty or not
            if(v5.value!=="")
            {
                f4=true;
            }
            else
            {
                alert("Email is Empty...");
                f4=false;
            }
        // if all the flags are true then display the success and print the values in same page
            if(f1&&f2&&f3&&f4)
            {
                r1="Acount Name: "+v2.value+"\n";
                r2="Aadhaar Number: "+v3.value+"\n";
                r3="Mobile Number: "+v4.value+"\n";
                r4="Email Id: "+v5.value+"\n";
                alert("Success...\nData is verified...\n"+r1+r2+r3+r4);
            }
        }
    </script>
</head>
<body>
    <form name="fs" action="" method="post" onsubmit="disp()">
        <h3>User Registration System</h3><br>
```

```
        <div class="dd">
            Select the Type of Account:<br>
            <select id="ac">
                <option value="Savings Type">Savings</option>
                <option value="Current Type">Current</option>
            </select><br>
            <small></small><br>
            <input type="text" size="40" id="b1" placeholder="Account
Name"/><br>
            <small></small><br>
            <input type="text" size="40" id="b2" placeholder="Aadhar
Number"/><br>
            <small></small><br>
            <input type="tel" size="40" id="b3" placeholder="Mobile
Number"/><br>
            <small></small><br>
            <input type="email" size="40" id="b4" placeholder="Email
Address"/><br><br>
            <input type="submit" class="ss" value="Submit"/>
            <input type="reset" class="rr" value="Reset"/>
        </div>
    </form>
  </body>
</html>
```

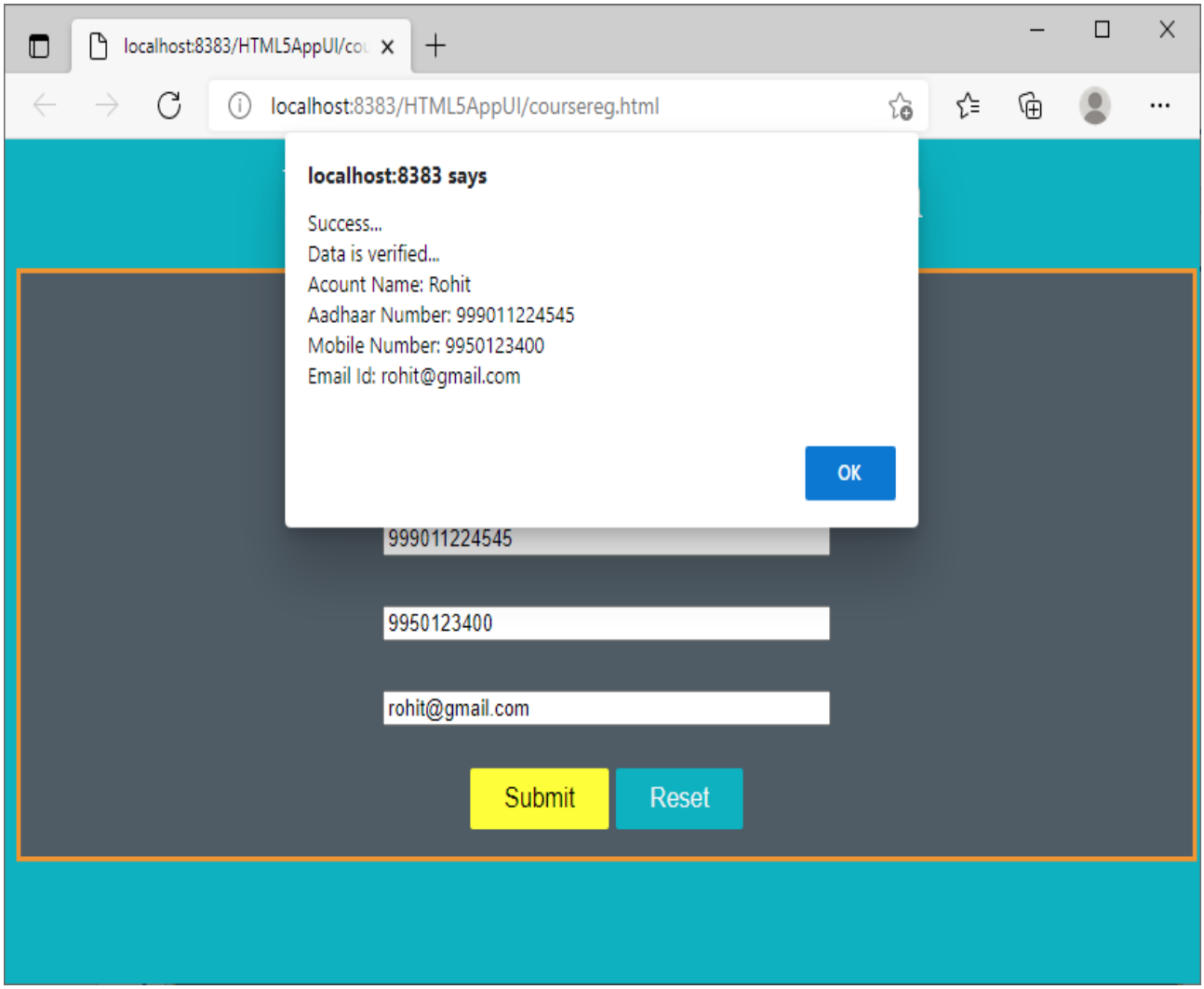
OUTPUT - HOME PAGE



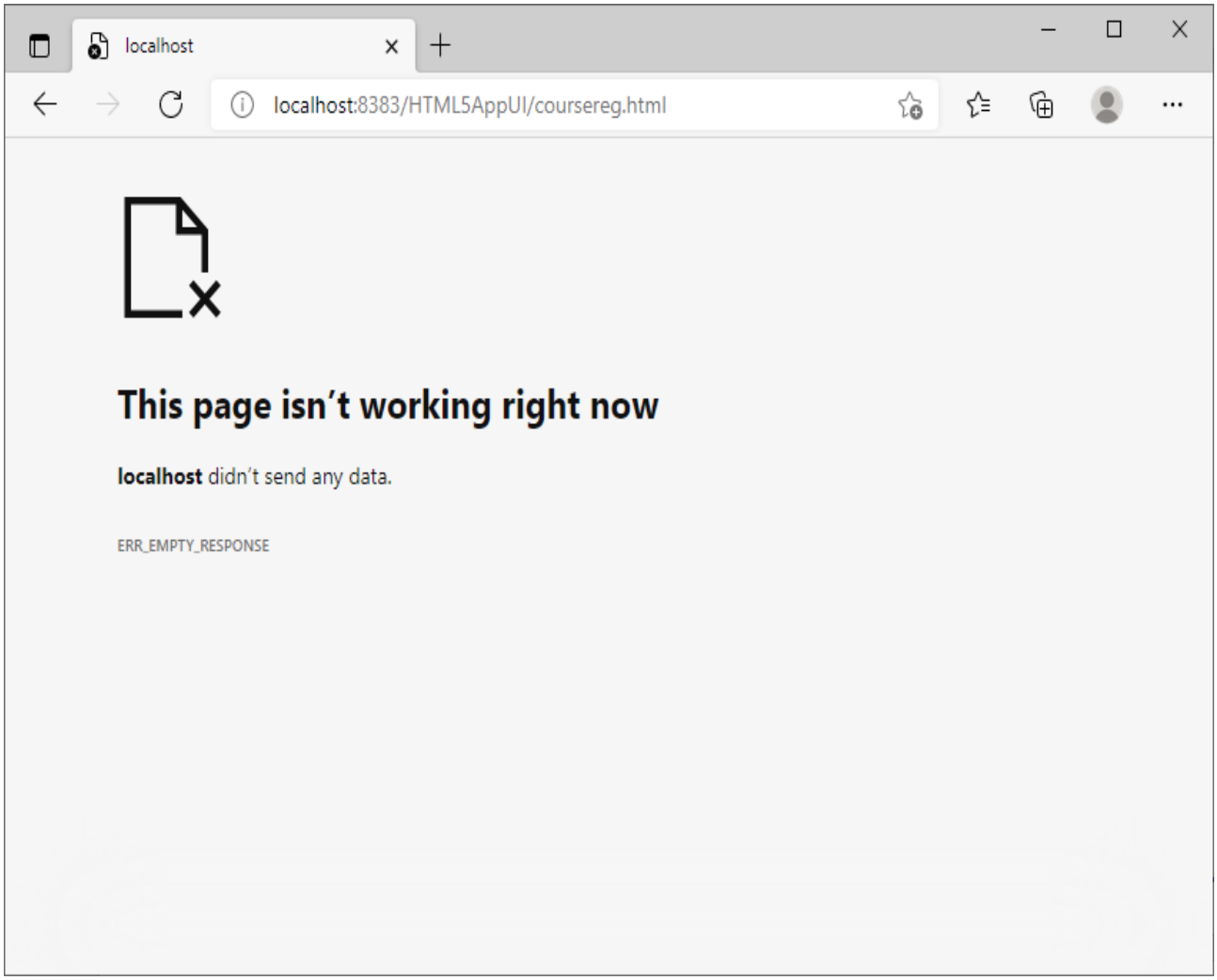
DATA SUBMISSION



SUCCESS CASE



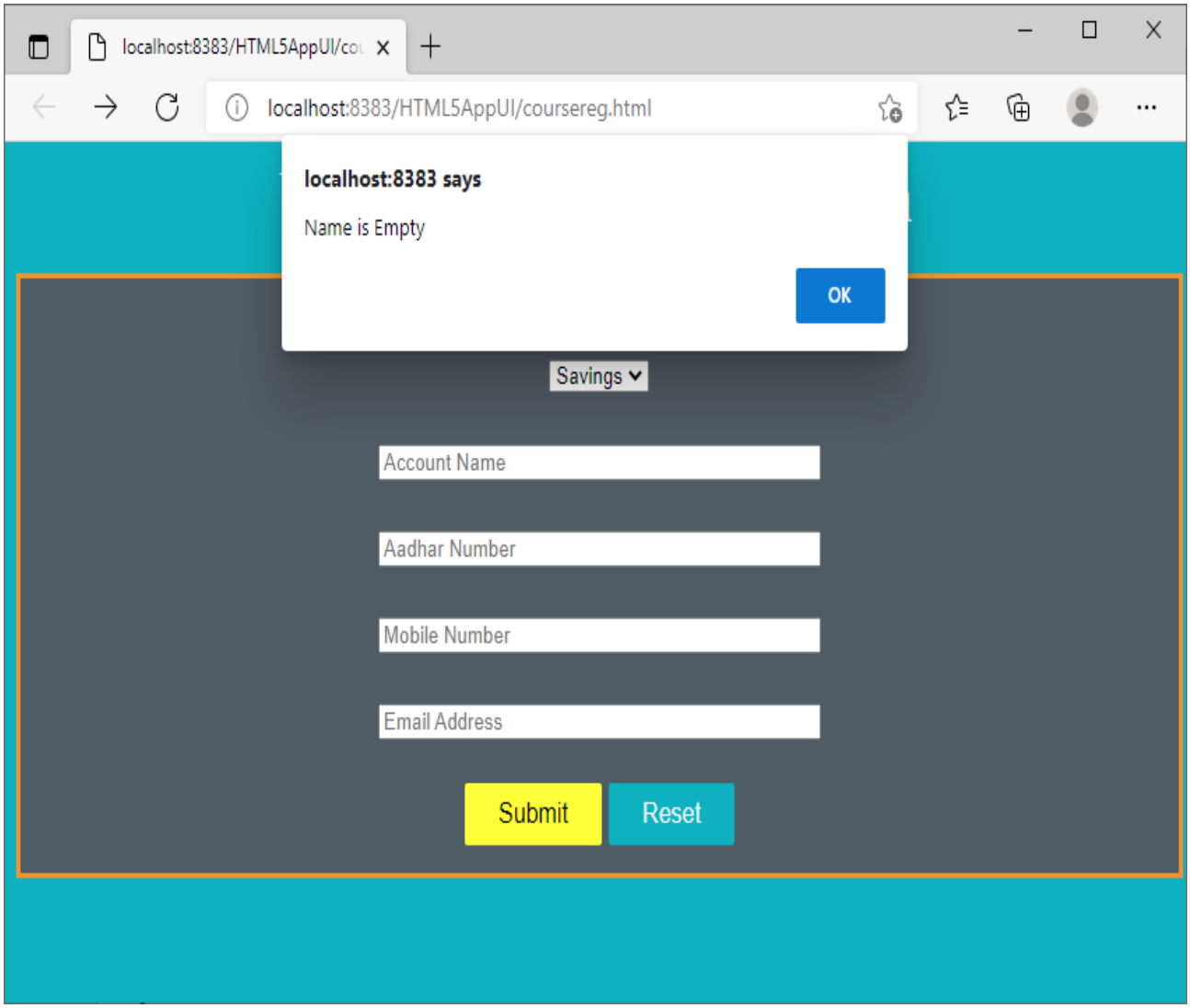
STATUS OF SERVER



NOTE

- The page above returns the server response like host didn't send any data due to not mentioned the server side languages like php, python, jsp, asp.net, etc,… in action attribute of form element.

FAILURE CASE



** tag (Grouping the HTML elements together into a section)**

- It stands for **Division Tag** and is called as content division element
- It is a special HTML tag which is used to organize the input html elements like button, textbox, label, etc, …
- It is a **container type** (not control type)
- It is mainly used to define a division or a section for HTML UI elements
- It is easily styled with help of CSS using its class attribute or id attribute.
- It is note that, by default browsers always place a line break before and after the element.

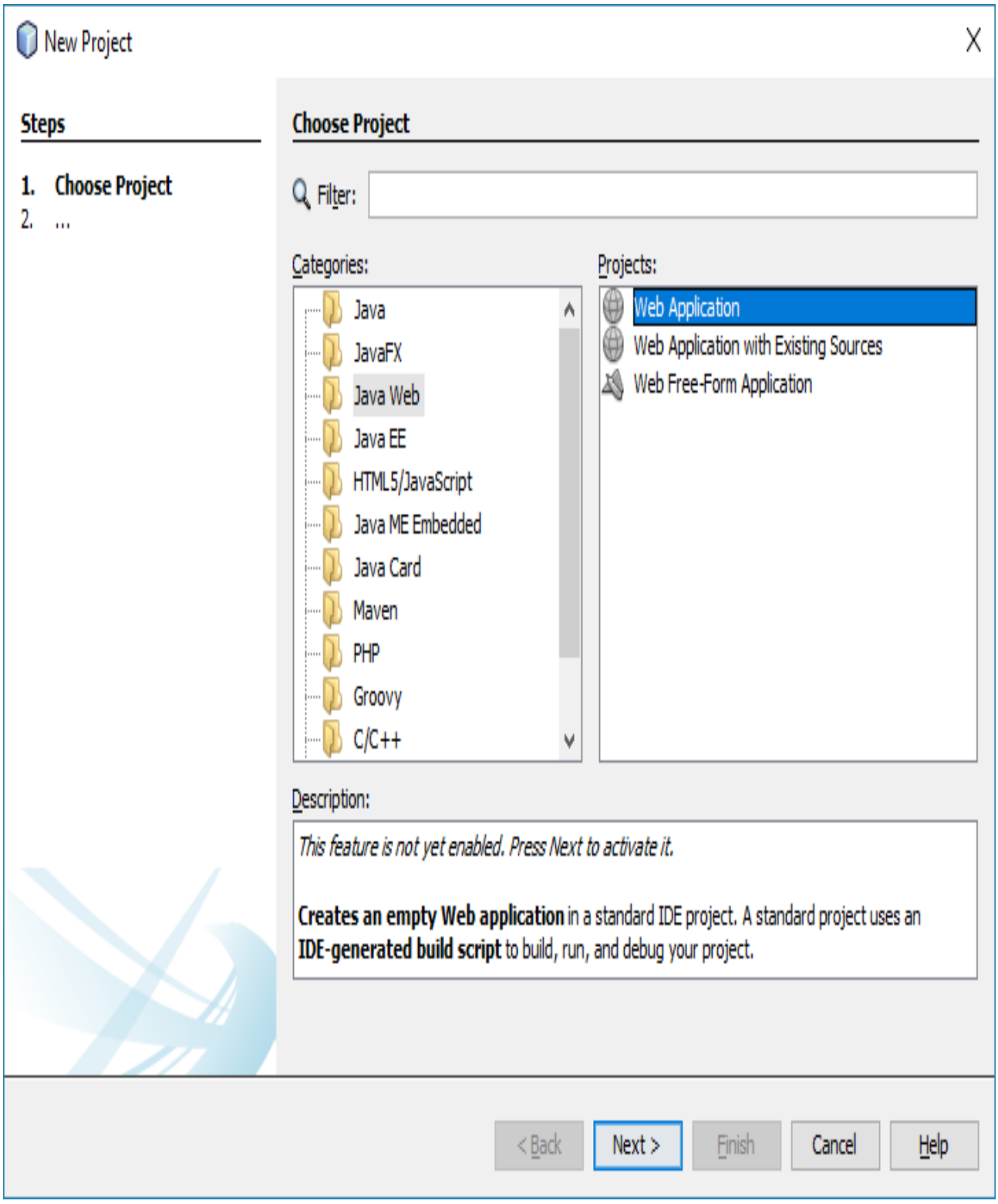
RESULT

Thus the creation of web form and validating it using java script was implemented successfully.

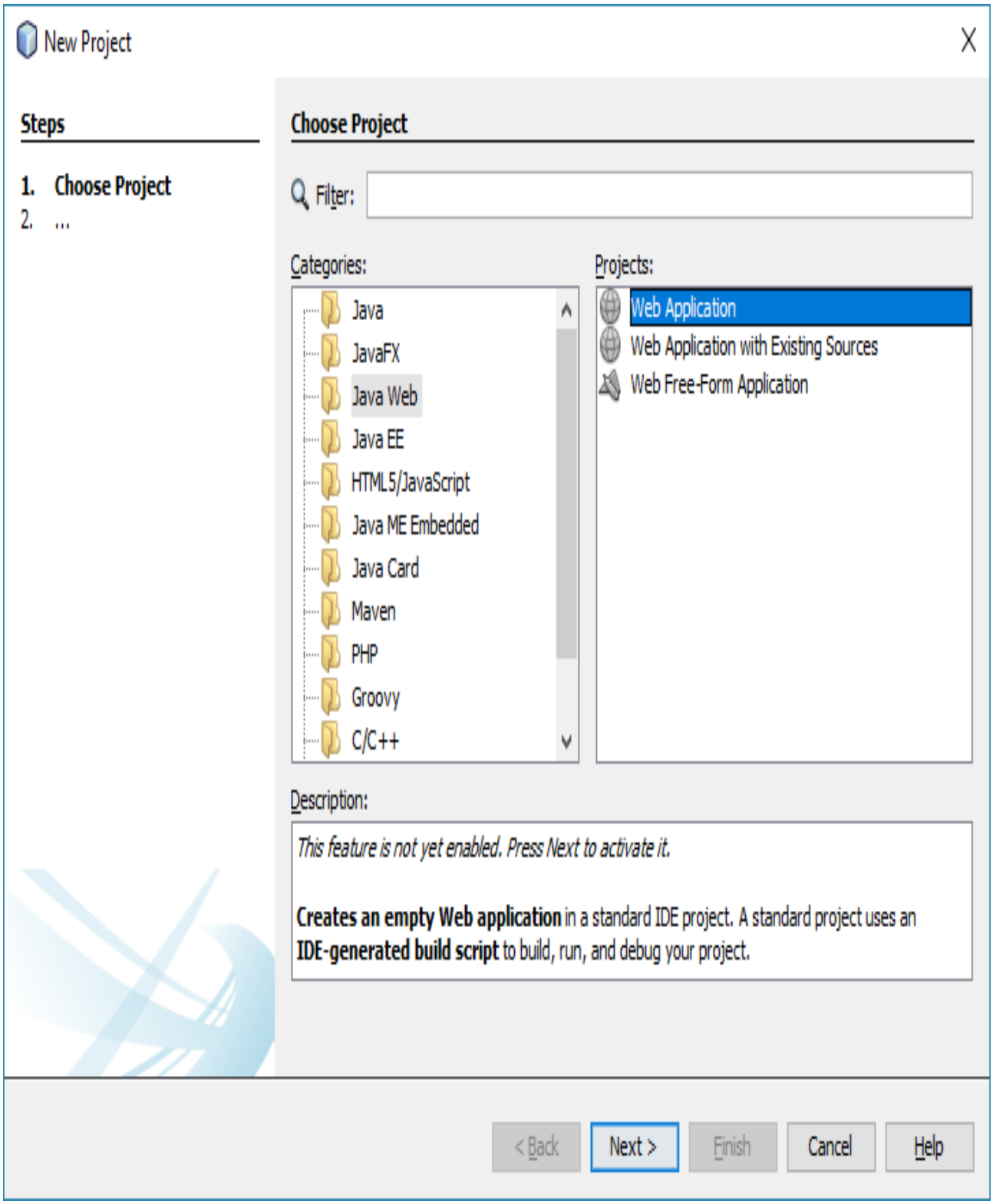
EX.NO: 12 **SIMPLE SERVLET IN JAVA**

I. CREATING HEELO WORLD USING SERVLET

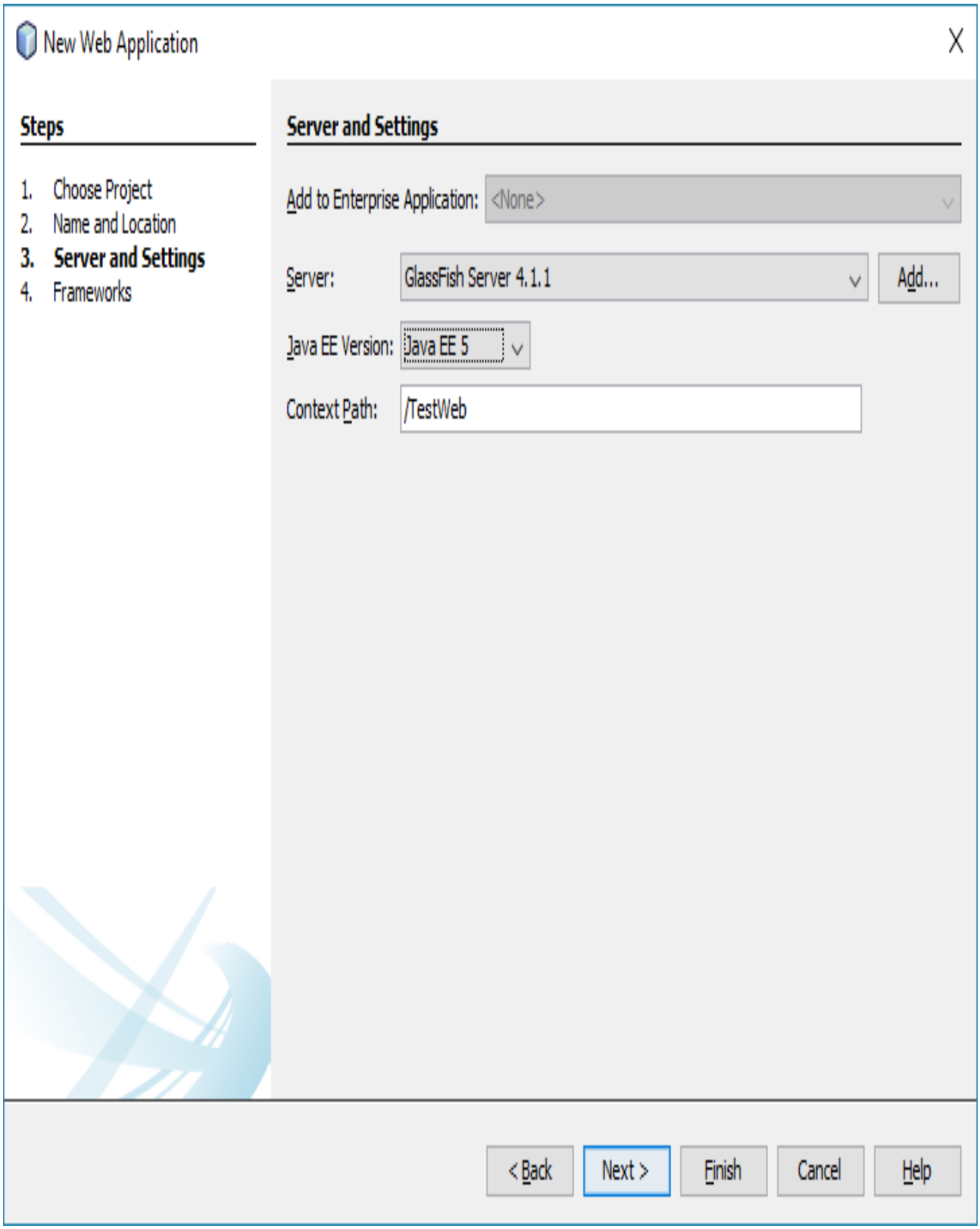
STEP 1. PROJECT CREATION



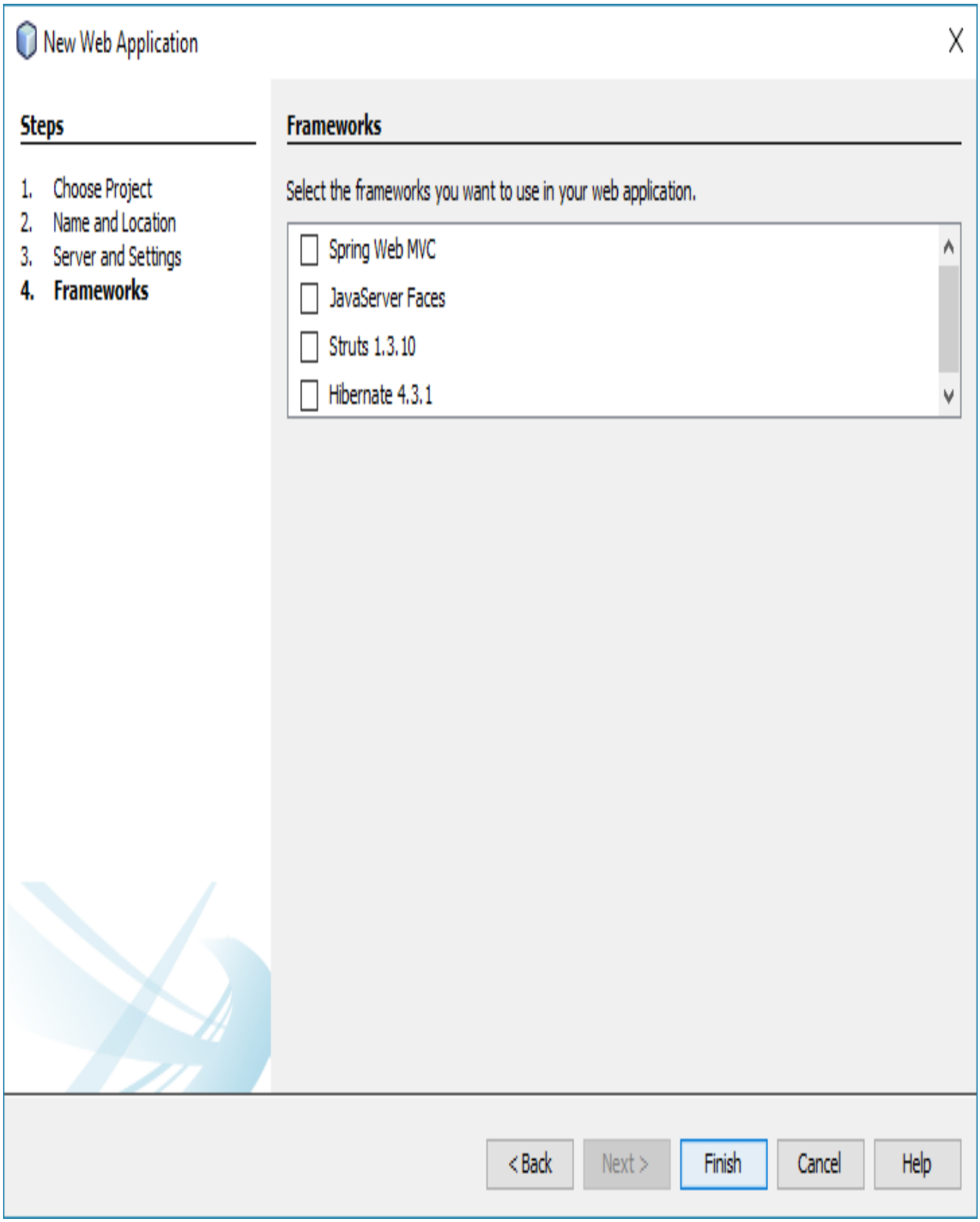
STEP 2. NAME AND LOCATION



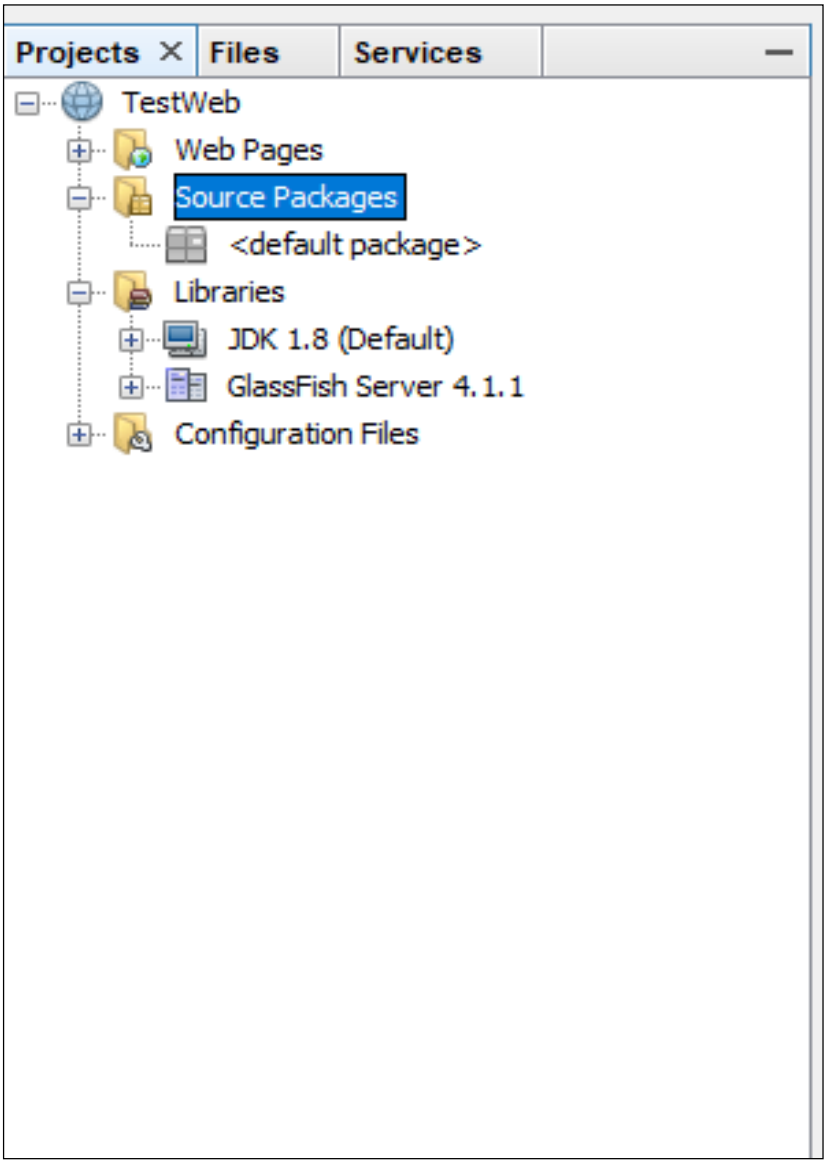
STEP 3. SERVER AND SETTINGS



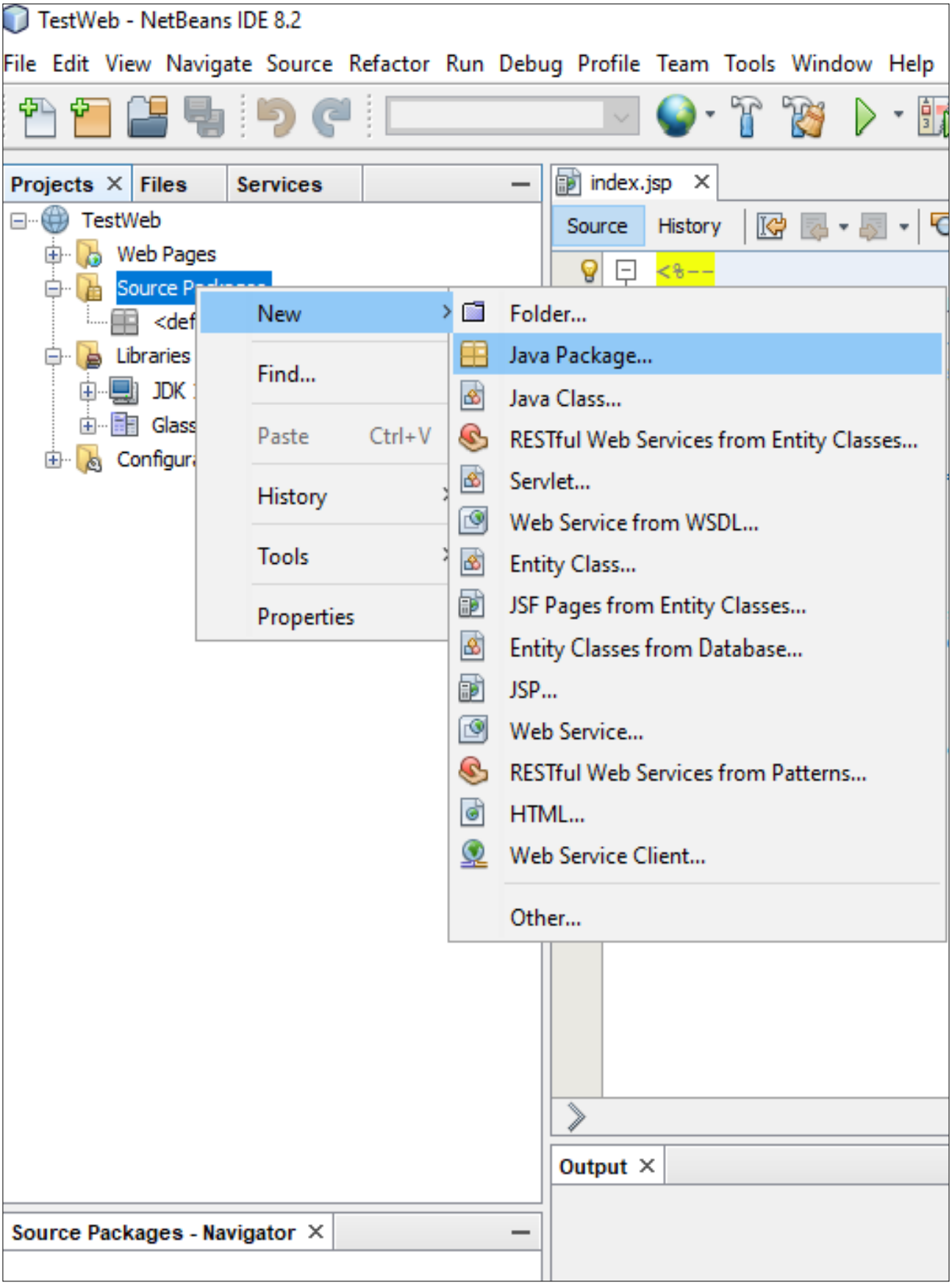
STEP 4. FRAMEWORKS: SELECT NONE



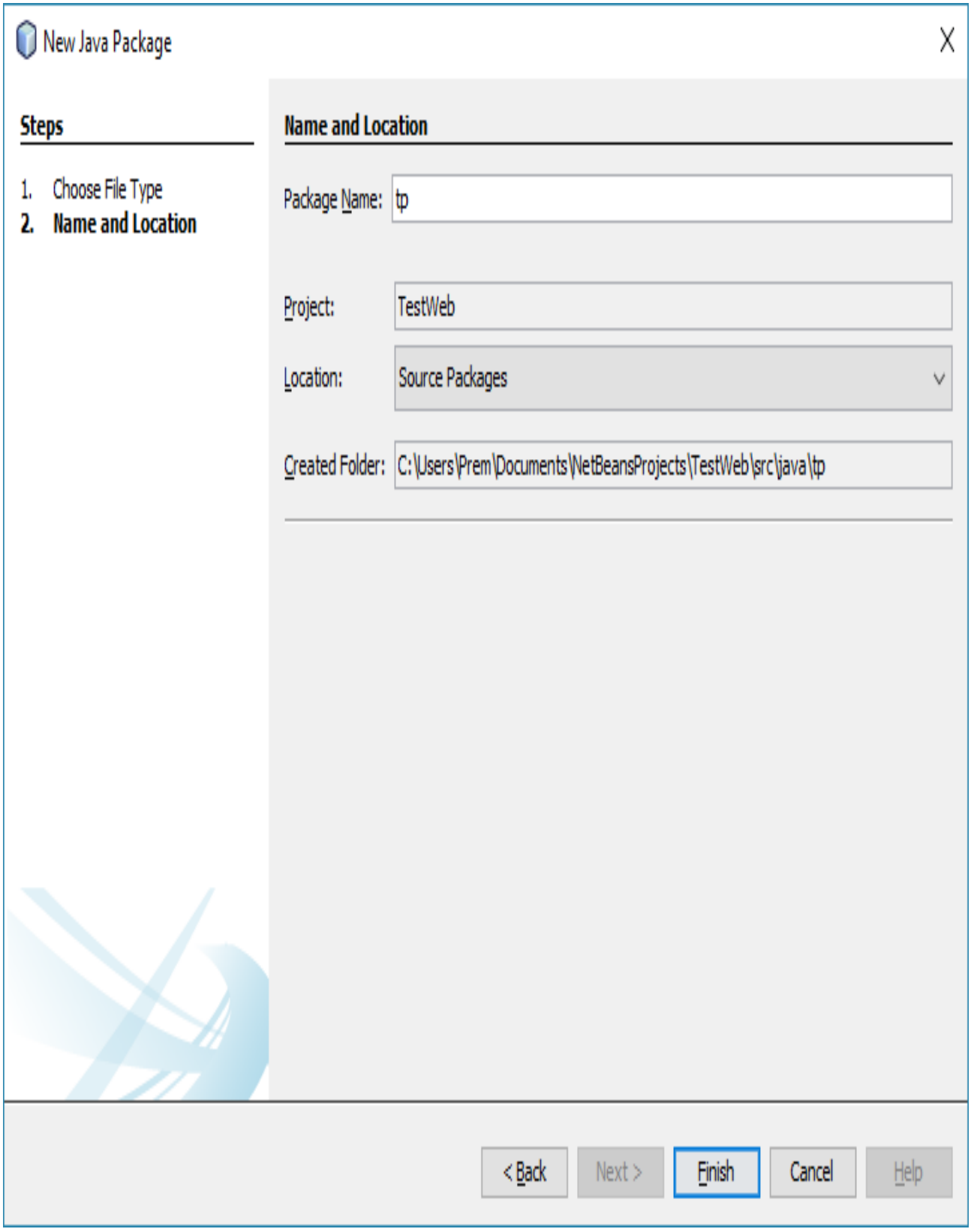
STEP 5. PROJECT STRUCTURE



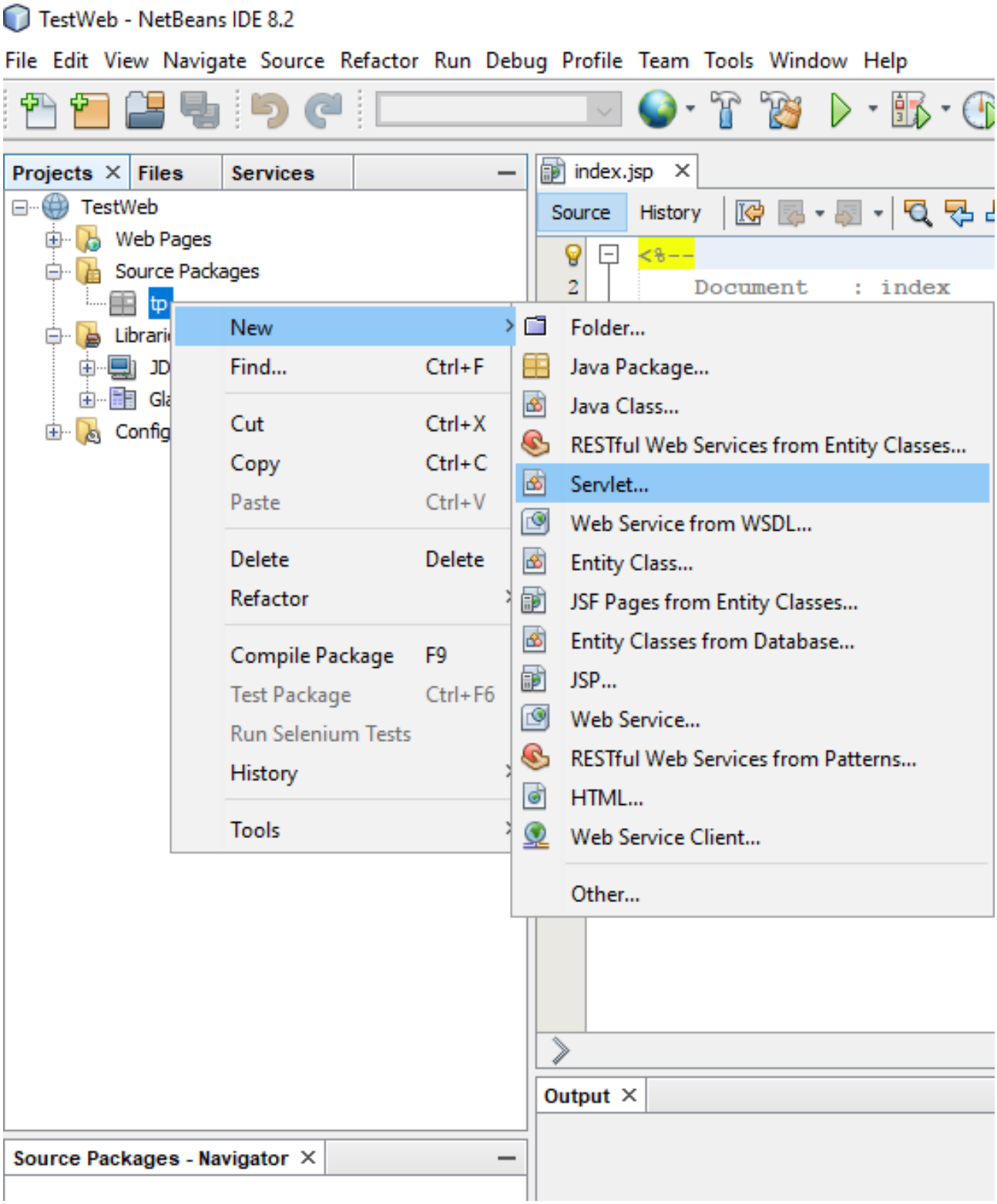
STEP 6. PROJECT CREATION



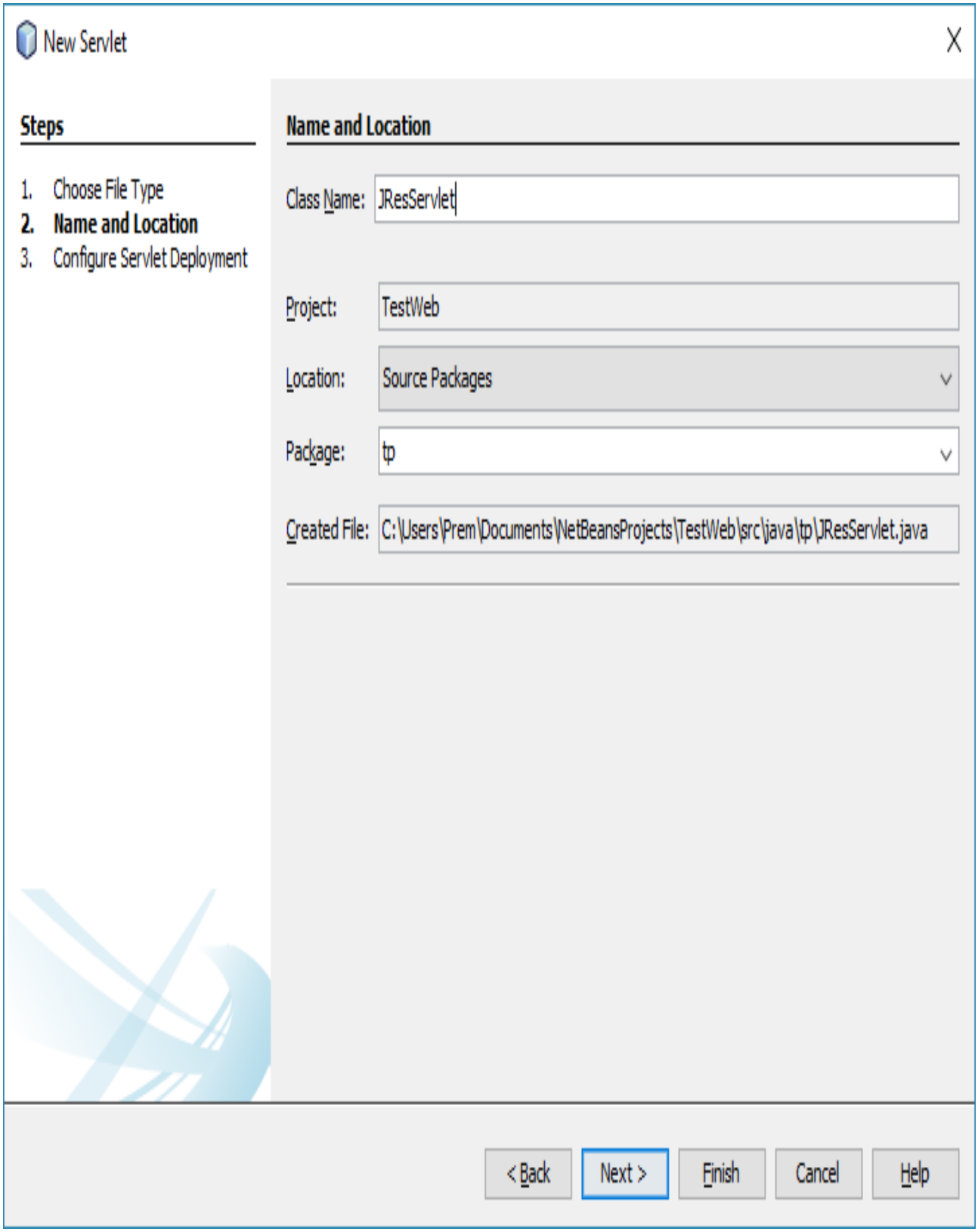
STEP 7. PROJECT CREATION (CONTINUE)



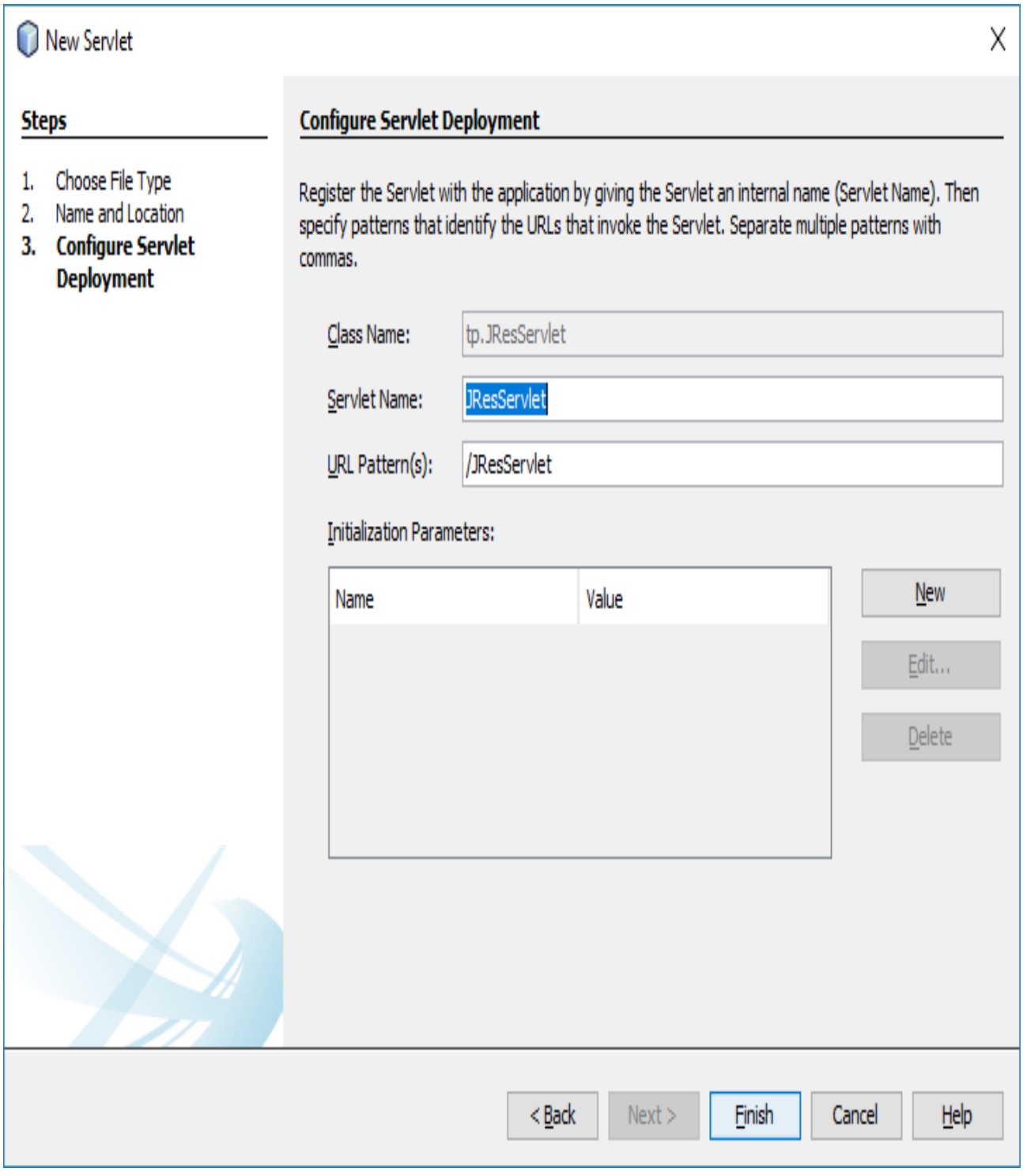
8. SERVLET CREATION UNDER CURRENT PACKAGE “tp”



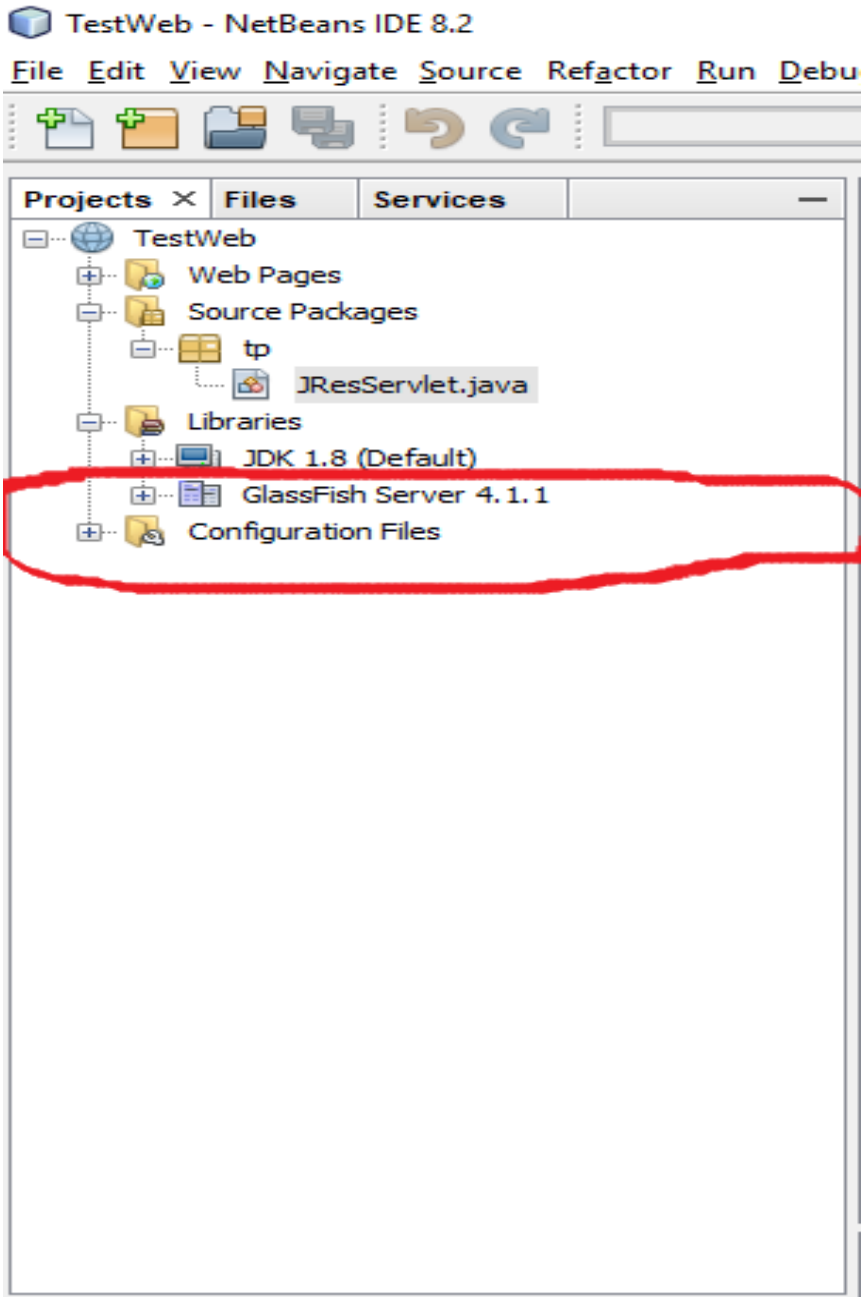
9. SERVLET CREATION UNDER CURRENT PACKAGE “tp” (CONTINUE)



10. CONFIGURE SERVLET DEPLOYMENT



11. COMPLETE PROJECT DETAILS



SOURCE CODE

(a) JSP FILE

(index.jsp)

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <body>
        <form action="JResServlet" method="post">
            Enter your name :<br/>
            <input type="text" name="t1"/><br/>
            <input type="submit" value="submit">
        </form>
    </body>
</html>
```

(b) SERVLET FILE

(JResServlet.java)

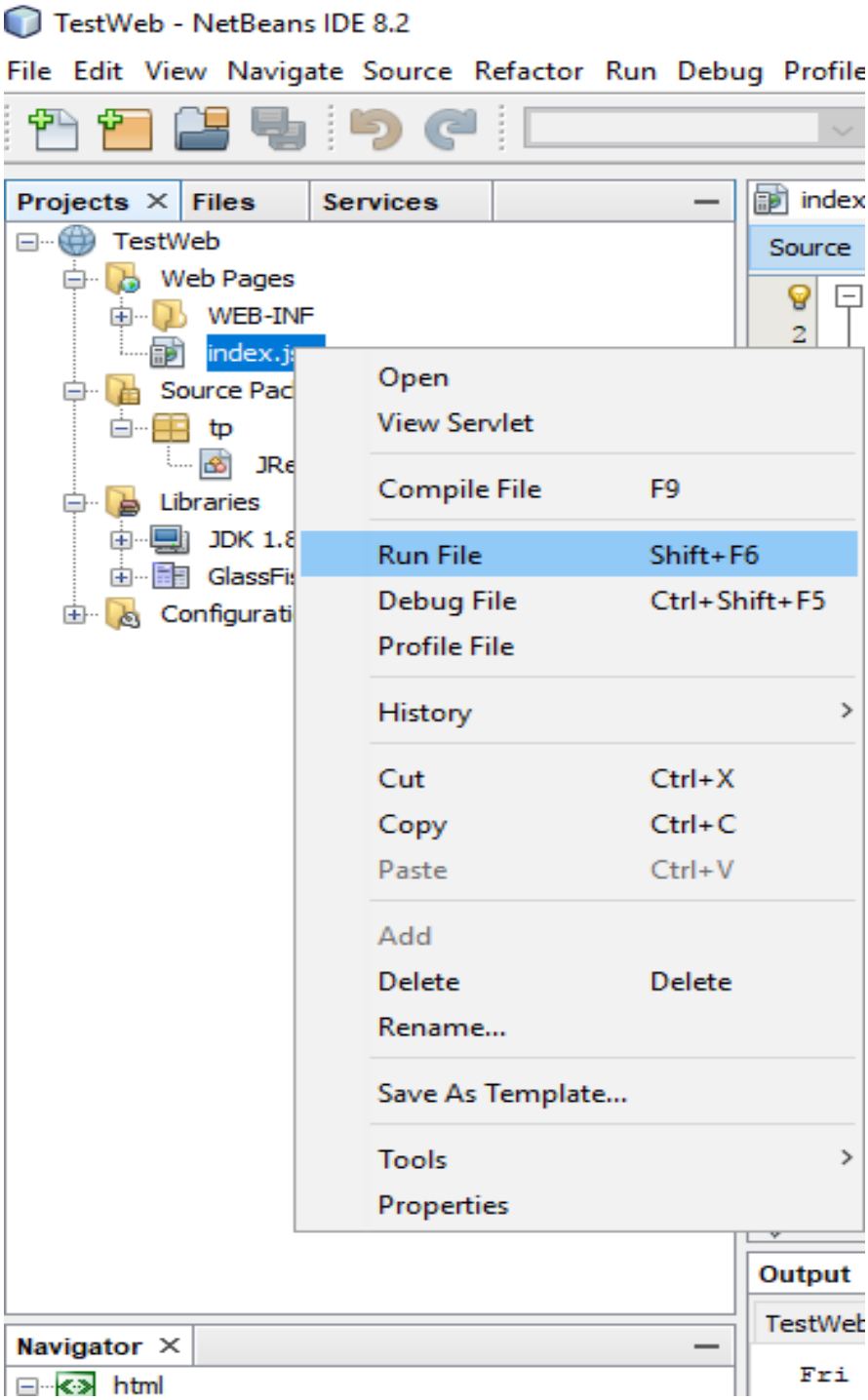
```
package tp;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class JResServlet extends HttpServlet {
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
```

```
        try (PrintWriter out = response.getWriter())
       {
        // get the input from client (JSP file)
            String name=request.getParameter("t1");
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet JResServlet</title>");
            out.println("</head>");
            out.println("<body>");
      // print owner name on client browser
            out.println("<h2>Ur Name is " + name + "</h2>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
```

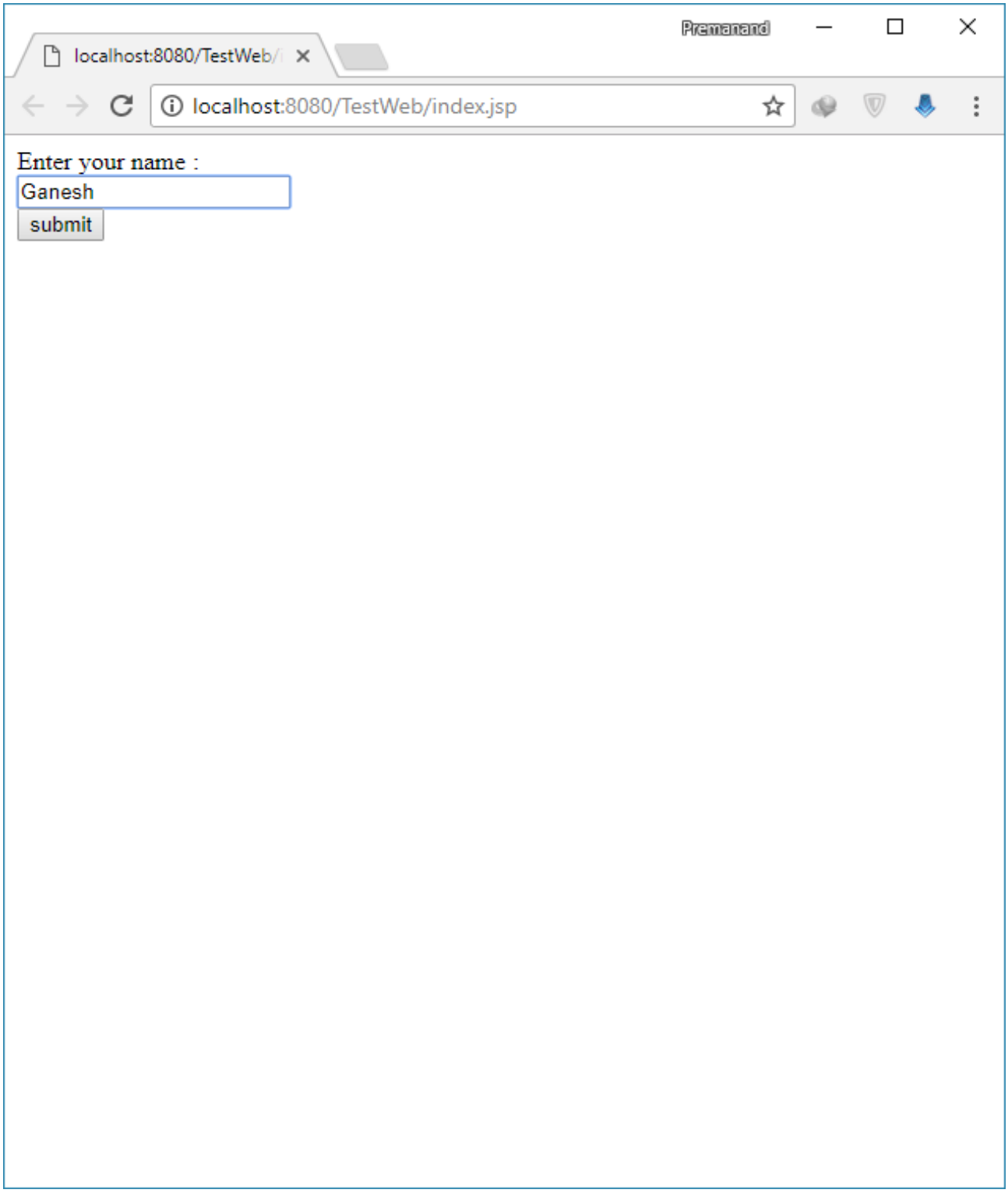
EXECUTION STEPS

1. Compile JSP file (.jsp)

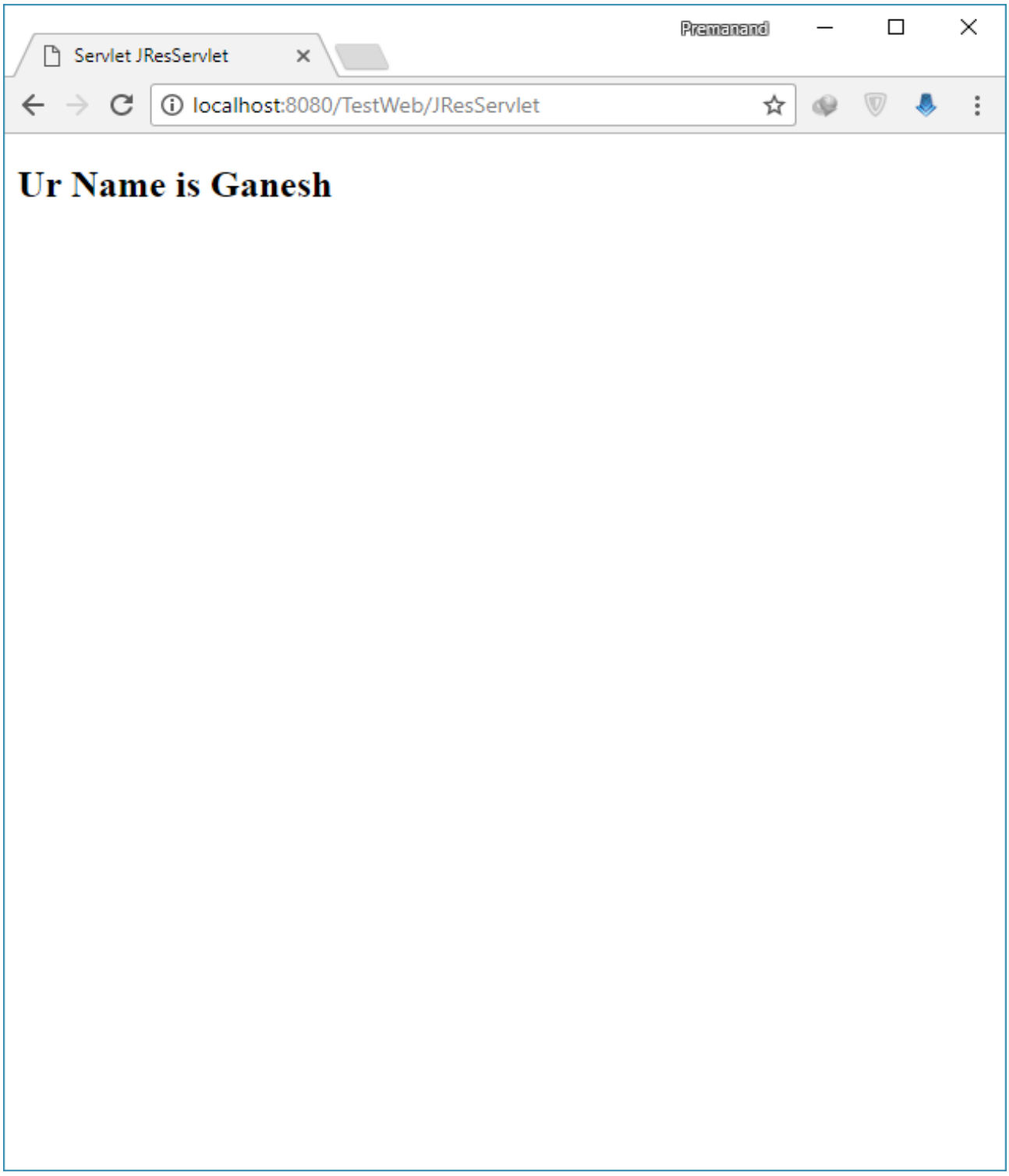
2. Run JSP file (.jsp)



OUTPUT - CLIENT REQUEST (USER INPUT QUERY)



SERVER RESPONSE (VIA SERVLET)



RESULT

Thus the servlet program was created and implemented using java successfully.

EX.NO: 13 SERVLET WEB BASED APPLICATION WITH JDBC

I. ONLINE EXAMINATION-3 TIER APPLICATION USING JSP AND SERVLET

SOURCE CODE

(Welcome.jsp)

```
<html>
<body>
// invoking servlet file in form element of JSP file
<form action="Scores" method="post">
<center><h1> Online Examination</h1><br>
UserName: <input type="text" name="t1"><br>
Seat Number: <input type="text" name="t2"><br>
Total marks:10(Each question carries equal marks)Time:
15mins.</center><br>
Q1.Which is pure oops?<br>
<input type="radio" name="g1" value="C++">C++
<input type="radio" name="g1" value="C">C
<input type="radio" name="g1" value="Shell">Shell
<input type="radio" name="g1" value="C#.NET">C#.NET<br>
Q2.Which is an example of compiled and interpreted language?<br>
<input type="radio" name="g2" value="C++">C++
<input type="radio" name="g2" value="C">C
<input type="radio" name="g2" value="Java">Java
<input type="radio" name="g2" value="HTML">HTML<br>
Q3.Which company developed java language?<br>
<input type="radio" name="g3" value="Microsoft">Microsoft
<input type="radio" name="g3" value="Apple">Apple
<input type="radio" name="g3" value="Google">Google
<input type="radio" name="g3" value="Sun">Sun<br>
Q4.Which company is very popular in selling smart phone in india?<br>
<input type="radio" name="g4" value="Samsung">Samsung
<input type="radio" name="g4" value="LG">LG
<input type="radio" name="g4" value="Nokia">Nokia
<input type="radio" name="g4" value="Sony">Sony<br>
Q5.Which team won the IPL2012?<br>
```

```
<input type="radio" name="g5" value="KKR">KKR
<input type="radio" name="g5" value="CSK">CSK
<input type="radio" name="g5" value="DD">DD
<input type="radio" name="g5" value="MIS">MIS<br>
<center><input type="submit" value="submit"></center>
</form>
</body>
</html>
```

Scores.java (Servlet code)
[Servlet code is used to give response to client via web page]

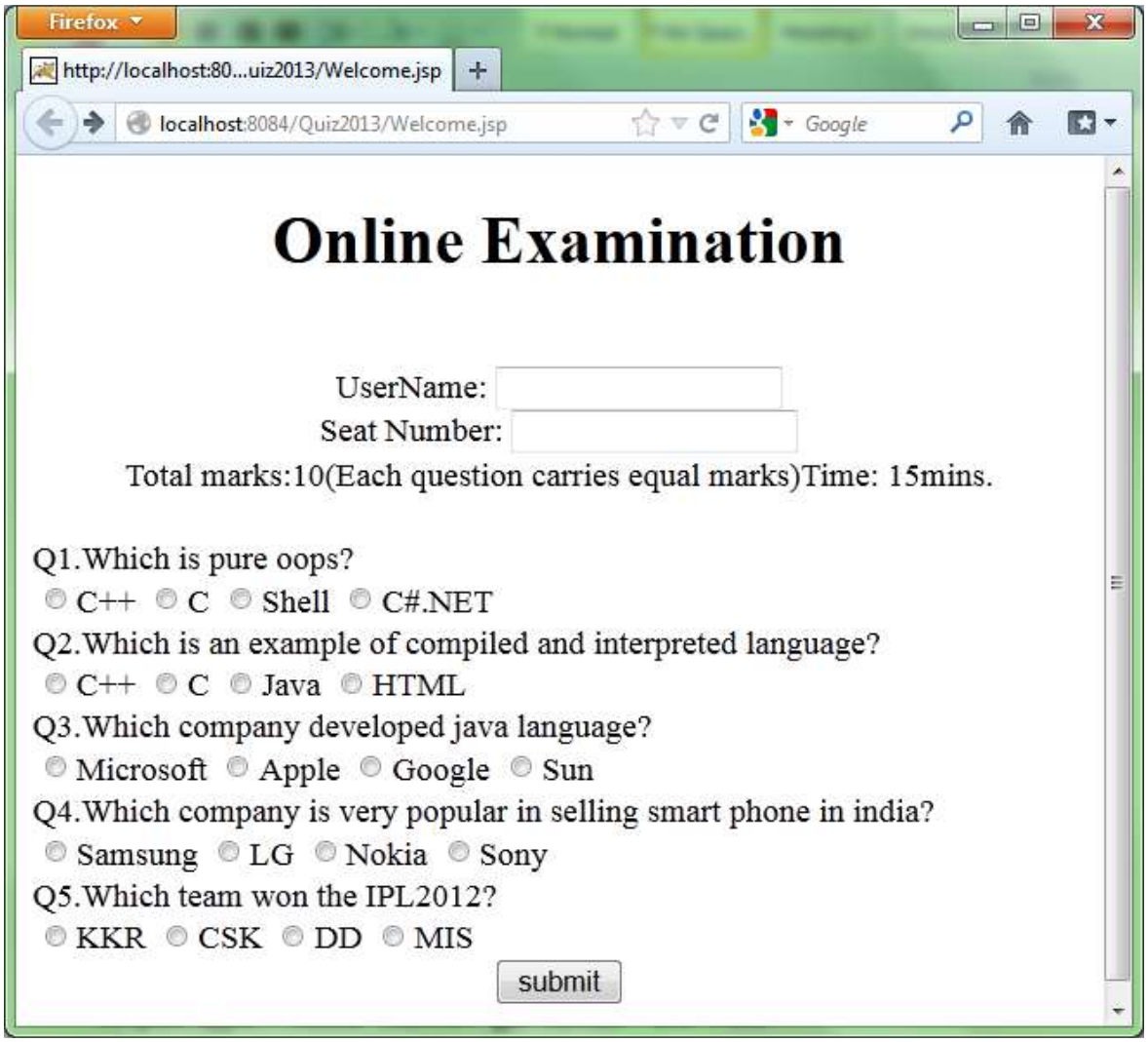
```
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
public class Scores extends HttpServlet {
Connection con;
PreparedStatement pt;
ResultSet rs;
int sn;
String uname;
int a1,a2,a3,a4,a5,total;
String isql="insert into Table1 values(?,?,?,?)";
static int c=1;
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
// establishing connection to database
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con=DriverManager.getConnection("jdbc:odbc:webtech");
//out.println("<h3>DB is connected...</h3>");
}
```

```
catch(Exception e)
{
out.println("<h3>DB is not yet connected...</h3>");
}
// gathering HTML parameters (User name & Seat number)
uname=request.getParameter("t1");
sn=Integer.parseInt(request.getParameter("t2"));
// handling 1st question
String ans1=request.getParameter("g1");
if(ans1.equals("C#.NET"))
a1=2;
else
a1=0;
// handling 2nd question
String ans2=request.getParameter("g2");
if(ans2.equals("Java"))
a2=2;
else
a2=0;
// handling 3rd question
String ans3=request.getParameter("g3");
if(ans3.equals("Sun"))
a3=2;
else
a3=0;
// handling 4th question
String ans4=request.getParameter("g4");
if(ans4.equals("Samsung"))
a4=2;
else
a4=0;
// handling 5th question
String ans5=request.getParameter("g5");
if(ans5.equals("KKR"))
a5=2;
```

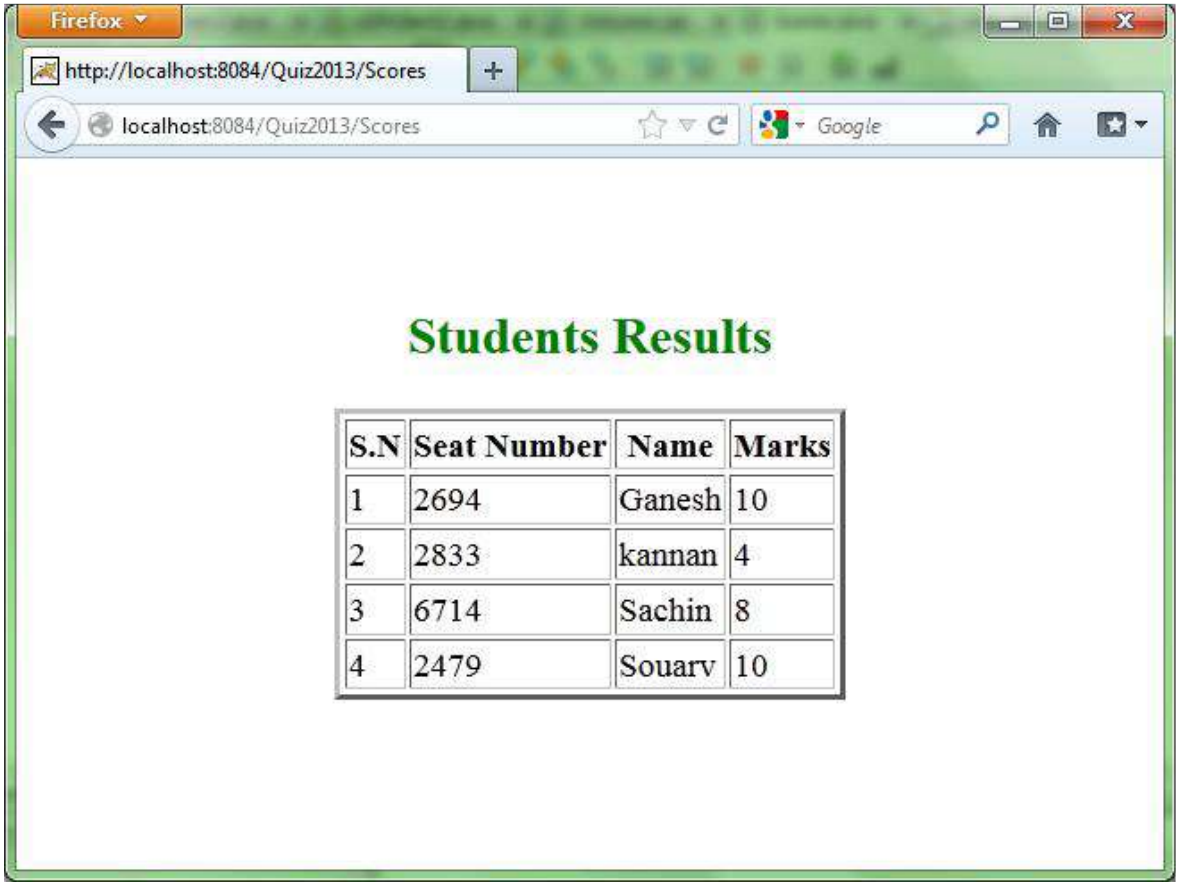
```
else
a5=0;
// collecting all answers
total=a1+a2+a3+a4+a5;
// inserting seat number, user name & total marks into database
try
{
pt=con.prepareStatement(isql);
pt.setInt(1, c++);
pt.setInt(2, sn);
pt.setString(3, uname);
pt.setInt(4,total);
int r=pt.executeUpdate();
pt.close();
}
catch(Exception e){}
// displaying the results from database
out.println("<html>");
out.println("<body>");
out.println("<center><br><br>");
out.println("<h2><font color=green>Students Results</h2>");
out.println("<table border=3>");
try {
pt=con.prepareStatement("select * from Table1");
rs=pt.executeQuery();
out.println("<th>S.N</th>");
out.println("<th>Seat Number</th>");
out.println("<th>Name</th>");
out.println("<th>Marks</th>");
while(rs.next())
{
out.println("<tr>");
out.print("<td>"+rs.getString(1));
out.print("<td>"+rs.getString(2));
out.print("<td>"+rs.getString(3));
```

```
out.print("<td>"+rs.getString(4));
out.println("</tr>");
}
}
catch(Exception e){}
finally {
try
{
con.close();
pt.close();
rs.close();
}
catch(Exception e){}
out.close();
}
out.println("</table>");
out.println("</body>");
out.println("</html>");
}
}
```

OUTPUT – HOME PAGE



EXAM RESULTS



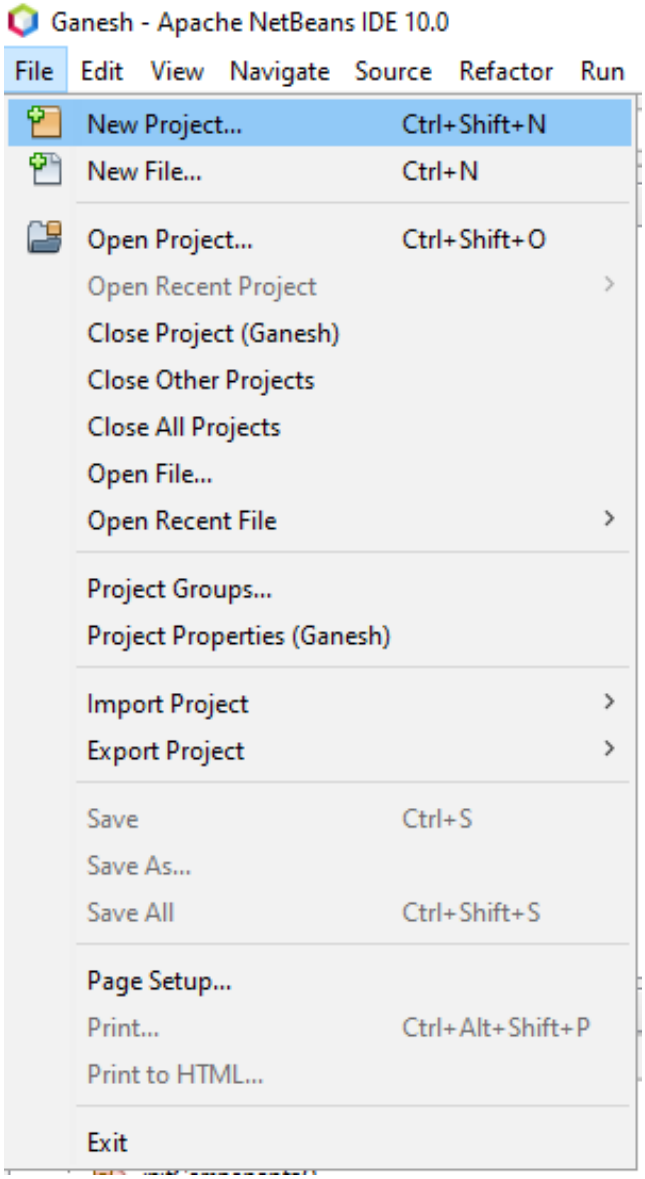
RESULT

Thus the servlet based web application with JDBC database was successfully implemented.

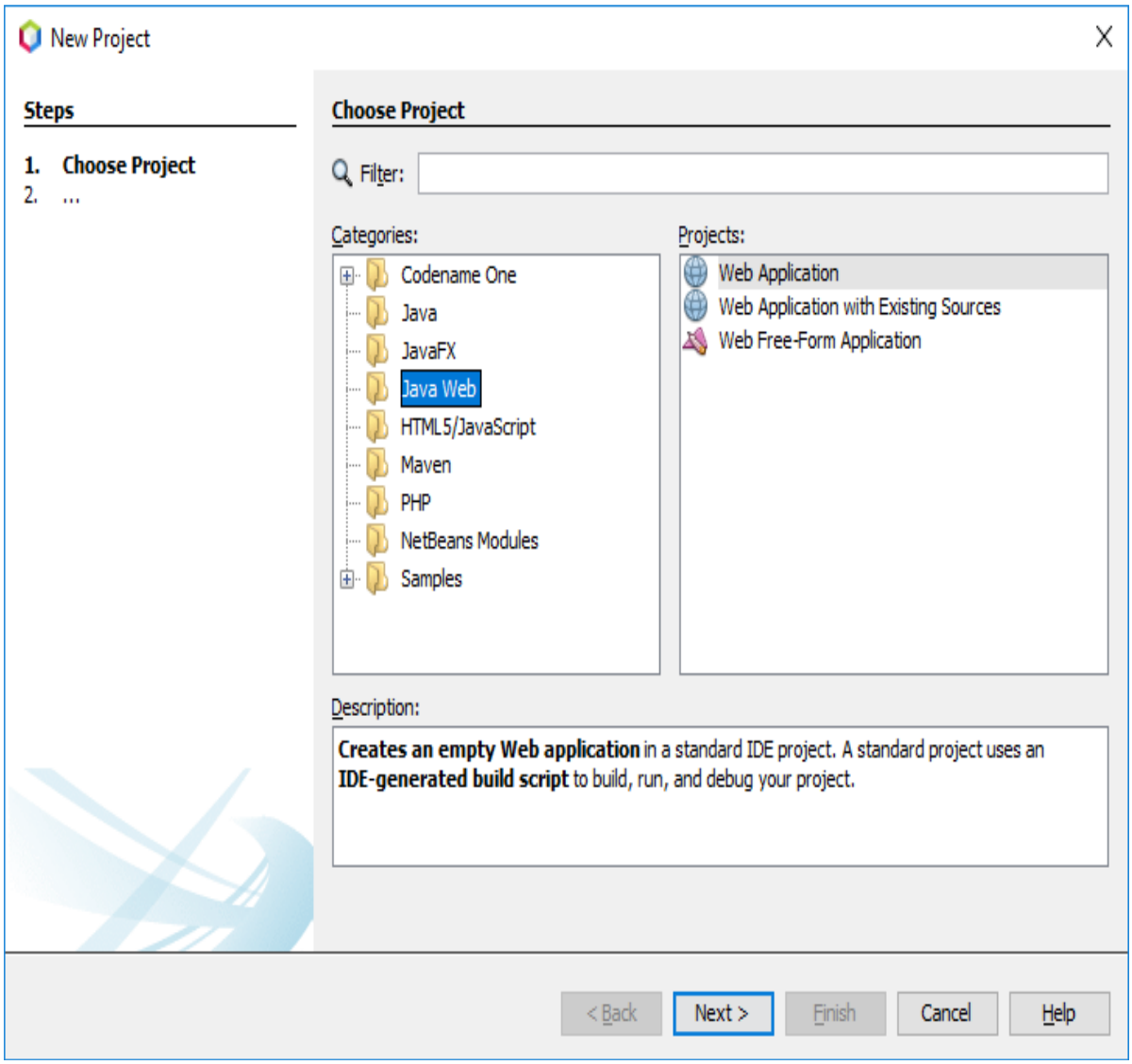
EX.NO: 14 DEVELOPING JSP APPLICATION

I. HELLO WORLD WEB APPLICATION USING JSP

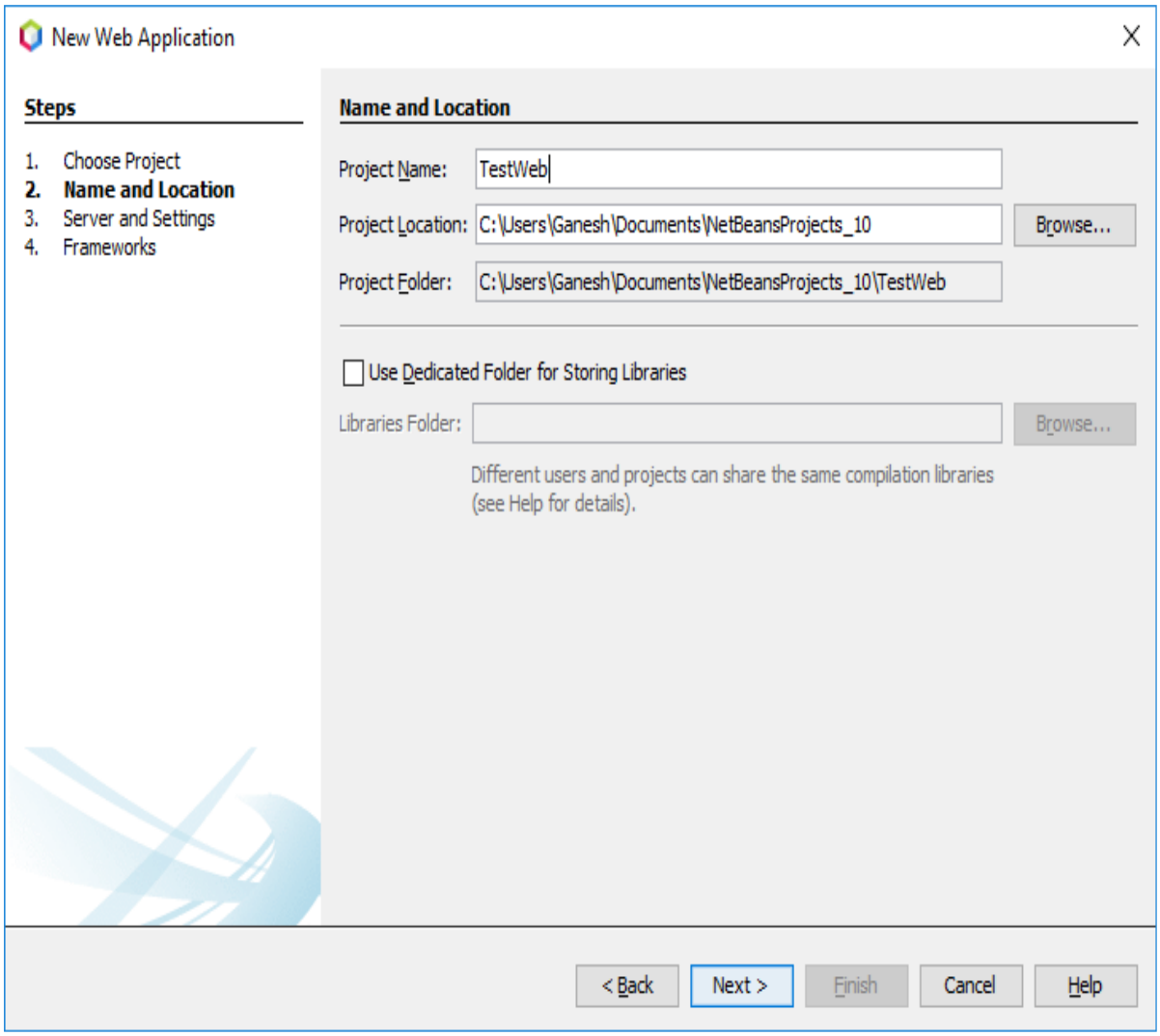
STEP 1: PROJECT CREATION (NETBEANS IDE 10.0)



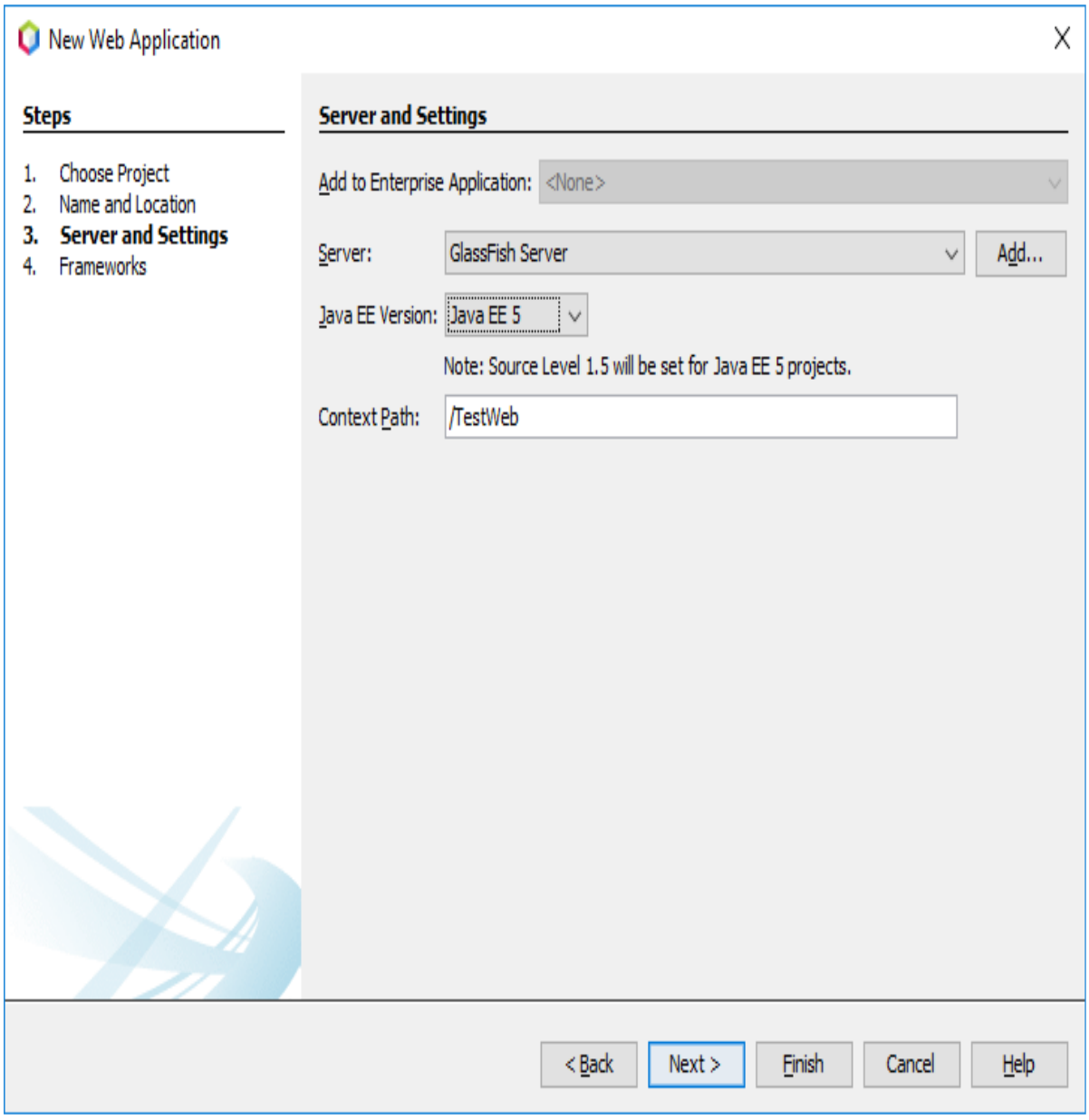
STEP 2: PROJECT CATEGORY SELECTION -> SELECT JAVA WEB



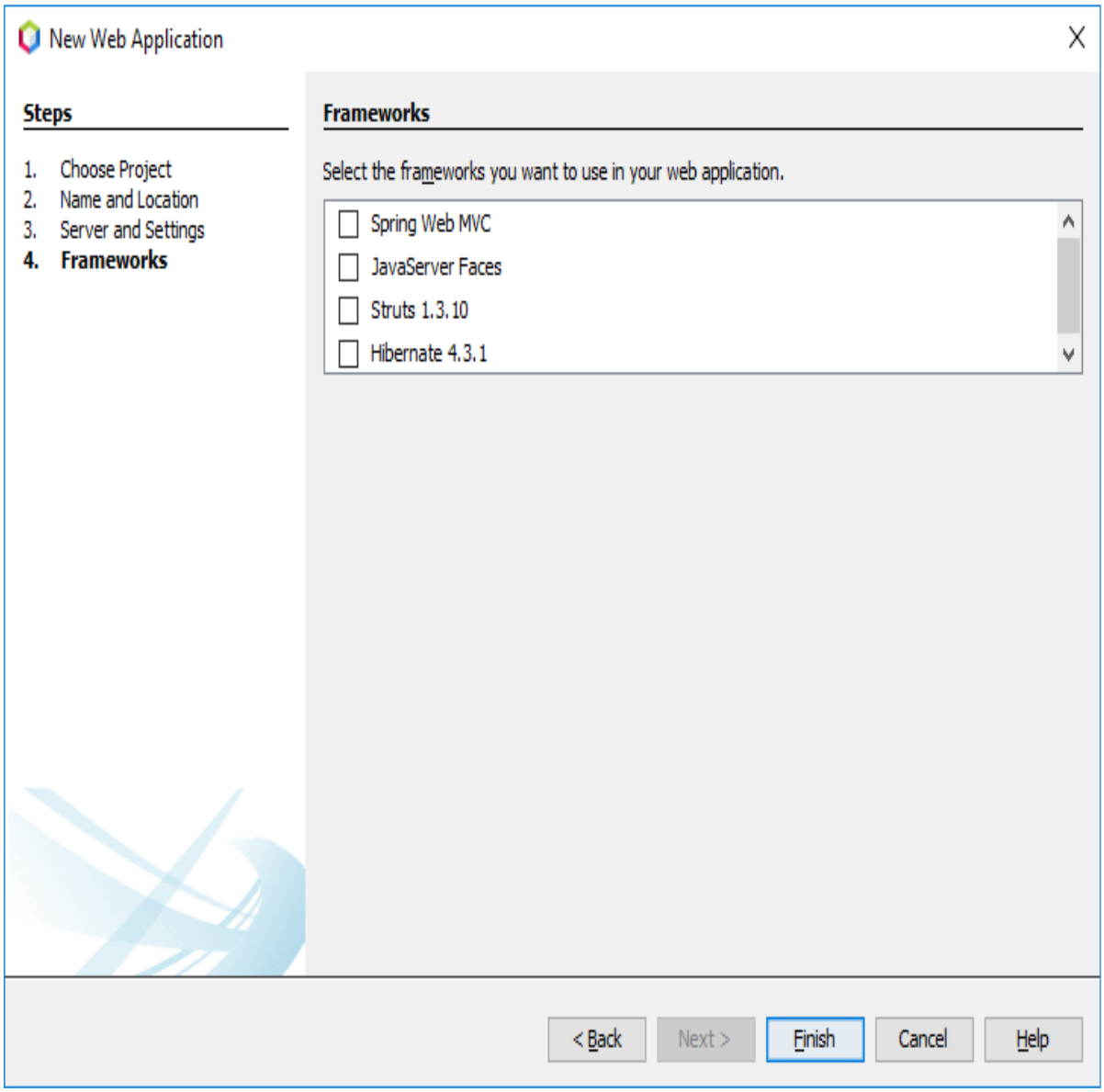
STEP 3: SELECTION OF THE PROJECT NAME AND ITS LOCATION



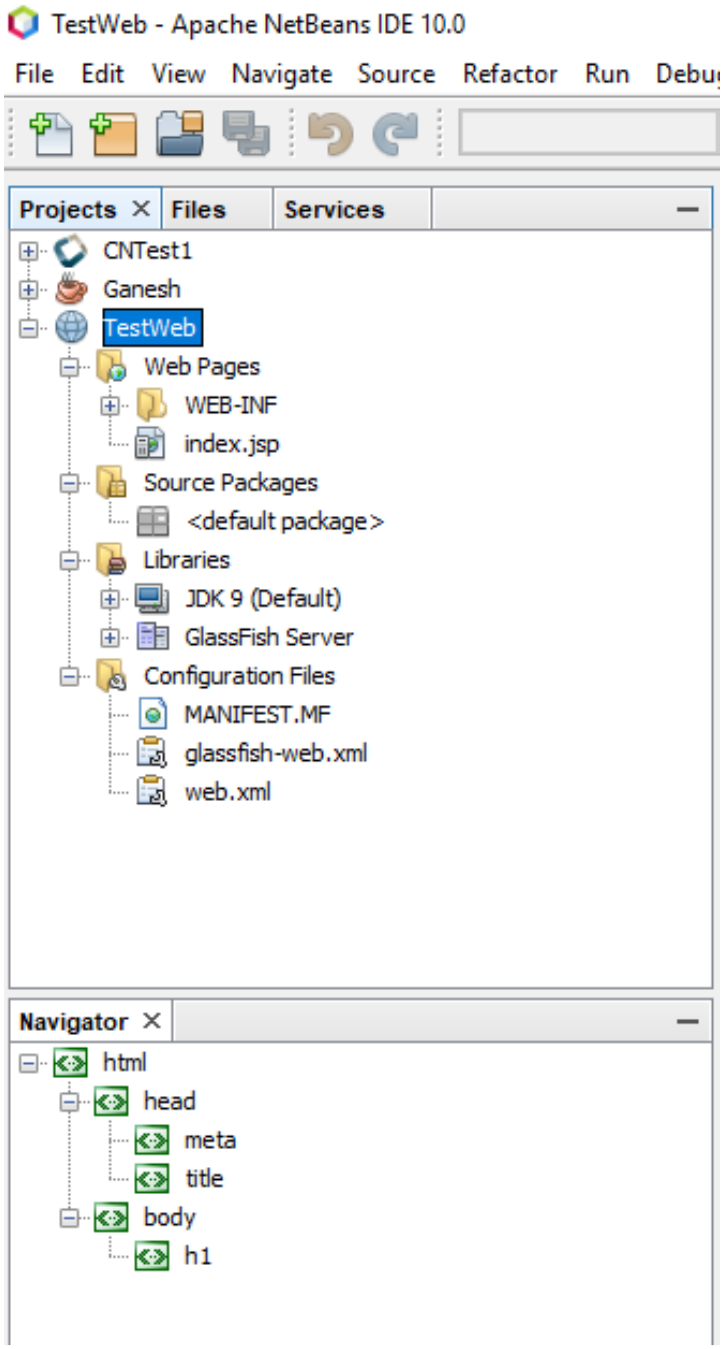
STEP 4: SERVER SELECTION AND SETTINGS (SELECTION OF GLASSFISH SERVER)



STEP 5: FRAMEWORK SELECTION (for advanced web application select any frameworks else uncheck the frameworks)



STEP 6: PROJECT STRUCTURE



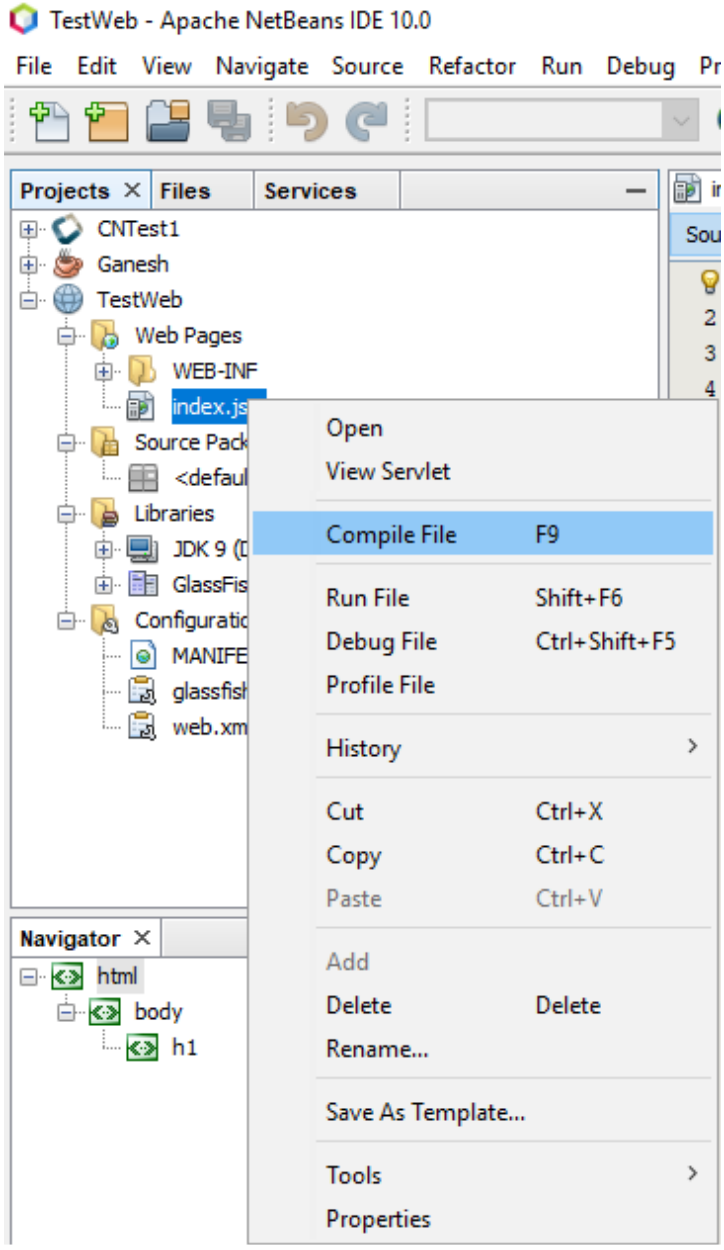
SOURCE CODE

(index.jsp)

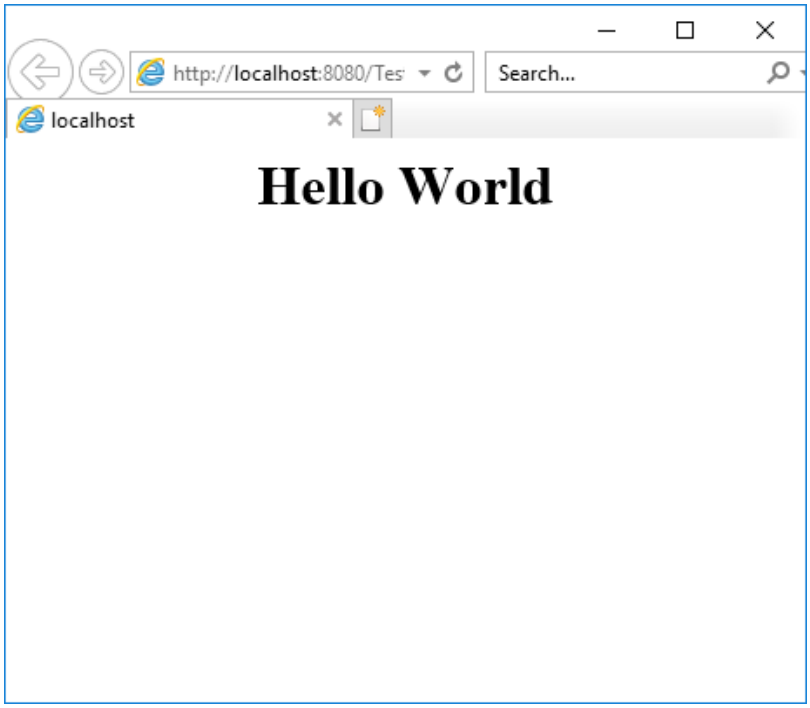
```
<html>
<body>
<center><h1>Hello World</h1></center>
</body>
</html>
```

STEP 7: COMPILE AND RUN JSP PAGE

- In order run web application, we need at least one server such as tomcat, glassfish, JBoss, etc, …
- Compile the target JSP file
- After successful compilation, run the current project by right click on current project and then select run option.



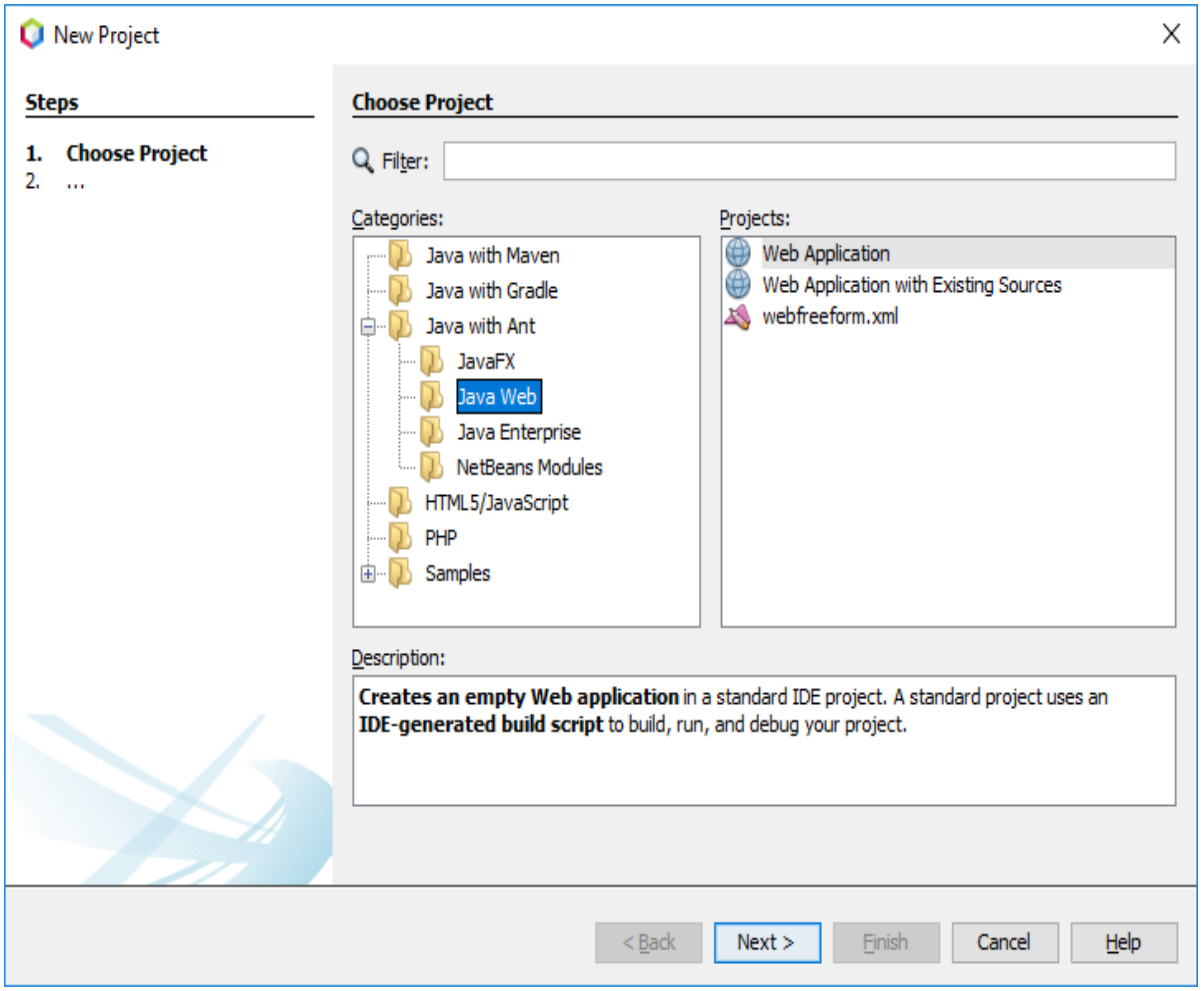
OUTPUT



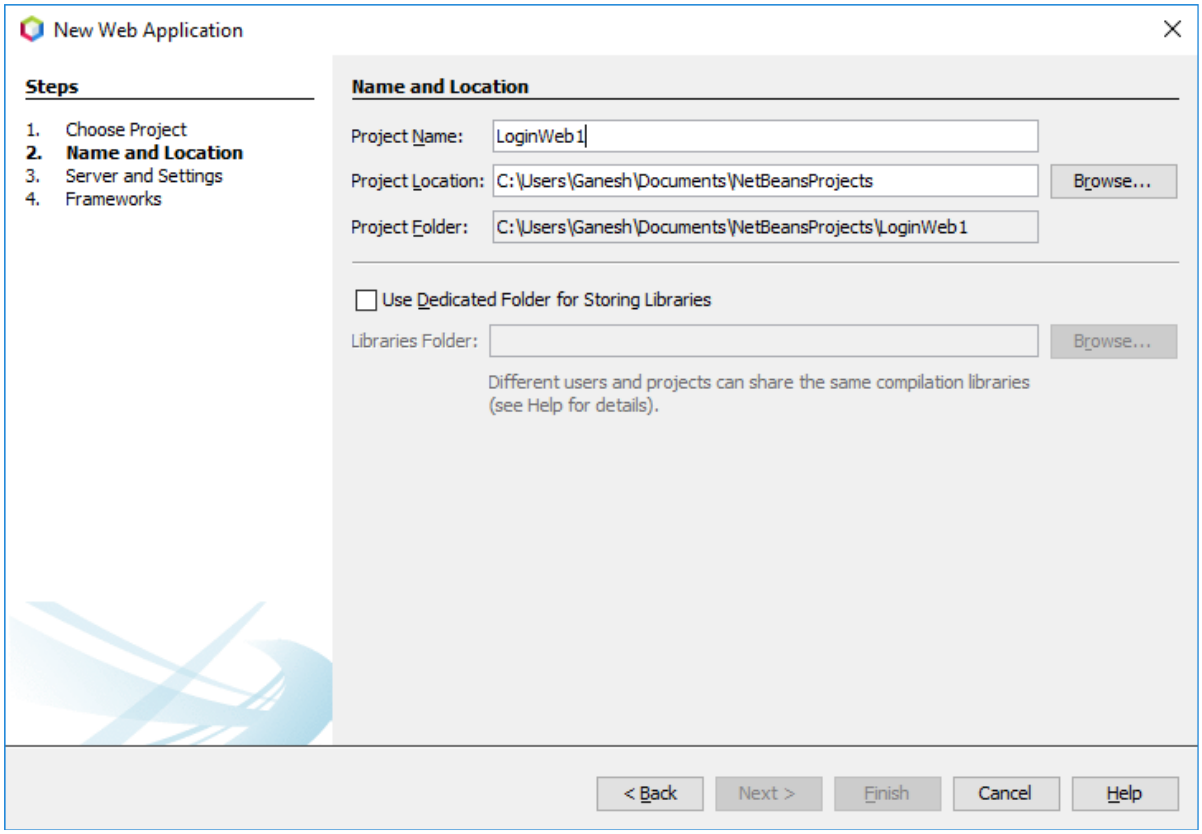
II. LOGIC FORM CREATION USING JSP WEB APPLICATION (CALLING JSP FROM HTML)

Application Type	:	Web Application
Front End	:	html (index.html)
Middle End	:	jsp (reply.jsp)
Tested IDE	:	Netbeans 11.3
Java EE Version	:	Java EE 8 Web
Web Server	:	Glassfish 5.0.1
Project Name	:	LoginWeb1
Context Path	:	/LoginWeb1
Output	:	HTML in Browser

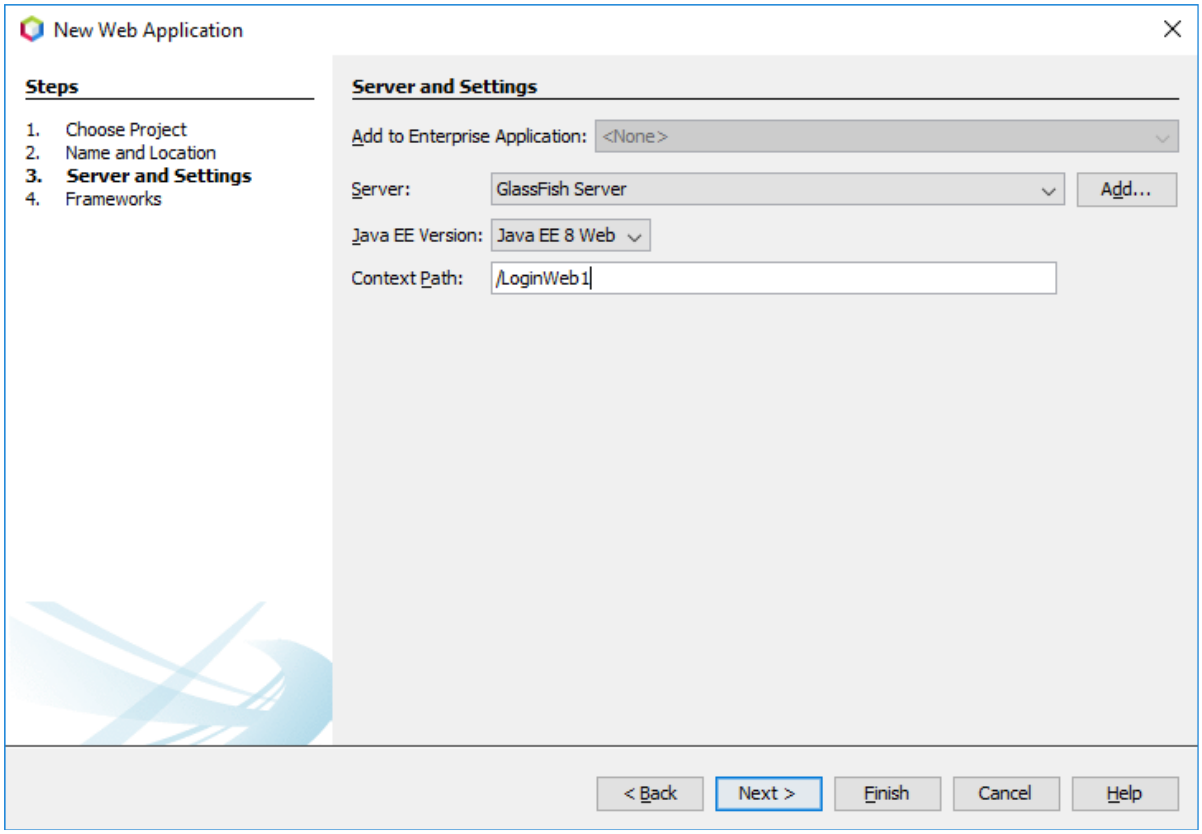
STEP 1: PROJECT SELECTION (NETBEANS IDE 11.3)



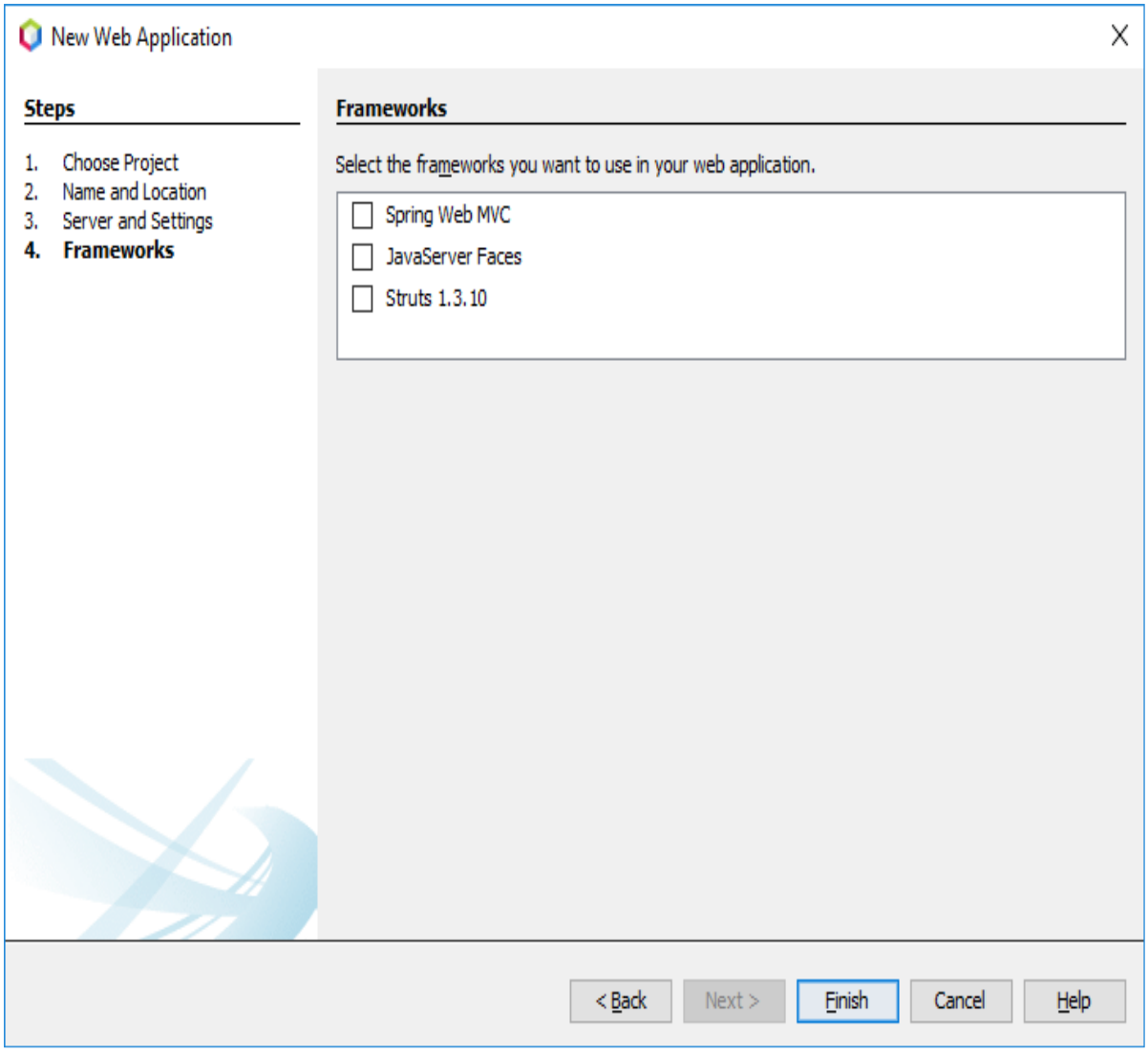
STEP 2: PROJECT SELECTION



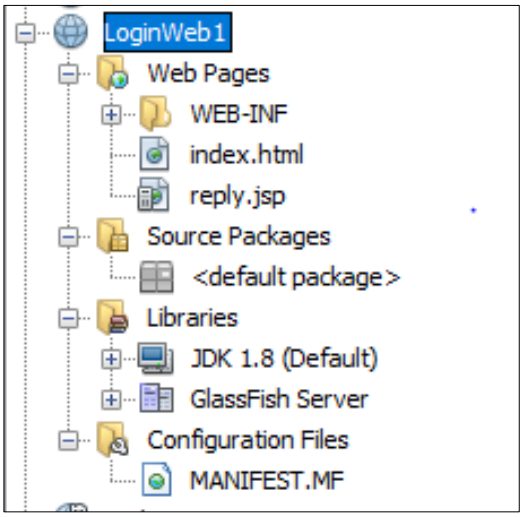
STEP 3: SERVER SELECTION AND J2EE SELECTION



STEP 4: FRAMEWORKS SELECTION



PROJECT STRUCTURE



SOURCE CODE

(index.html)

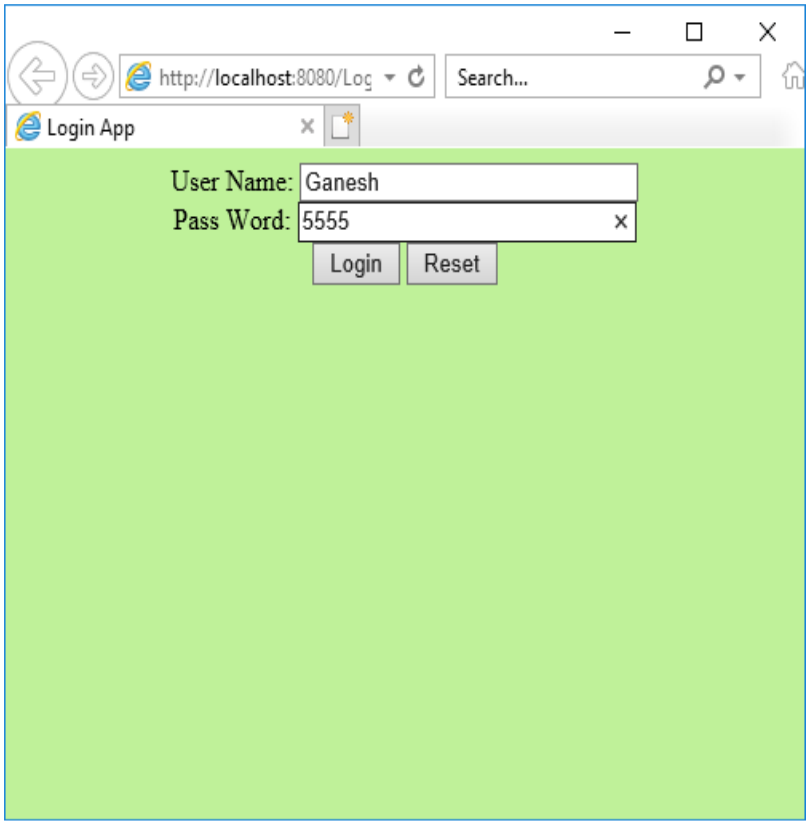
```
<html>
<head>
<title>Login App</title>
</head>
<body bgcolor="#bff199">
<center>
<form action="reply.jsp" method=”get”>
User Name: <input type="text" name="b1" size="30"/><br>
Pass Word: <input type="text" name="b2" size="30"/><br>
<input type="submit" value="Login"/>
<input type="reset" value="Reset"/>
</form>
</center>
</body>
</html>
```

(reply.jsp)

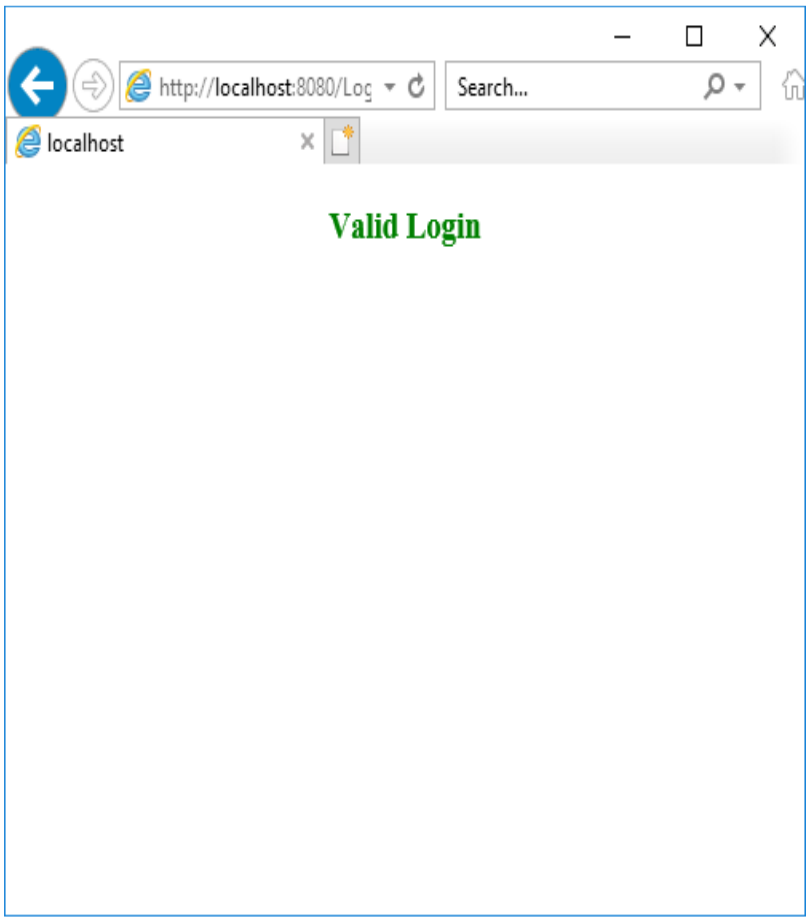
```
<html>
<body>
<%
int pw=0;
try
{
// handling client requests
// getting the 1st input from user
String user=request.getParameter("b1");
// getting the 2nd input from user
String pwd=request.getParameter("b2");
pw=Integer.parseInt(pwd);
// checking login details
if(user.equals("Ganesh")&&pw==5555)
{
out.println("<center><font color=green><h3>Valid
```

```
Login</h3></font></center>");
}
else
{
out.println("<center><font color=red><h3>Invalid
Login</h3></font></center>");
}
}
catch(Exception exp)
{
exp.printStackTrace();
}
%>
</body>
</html>
```

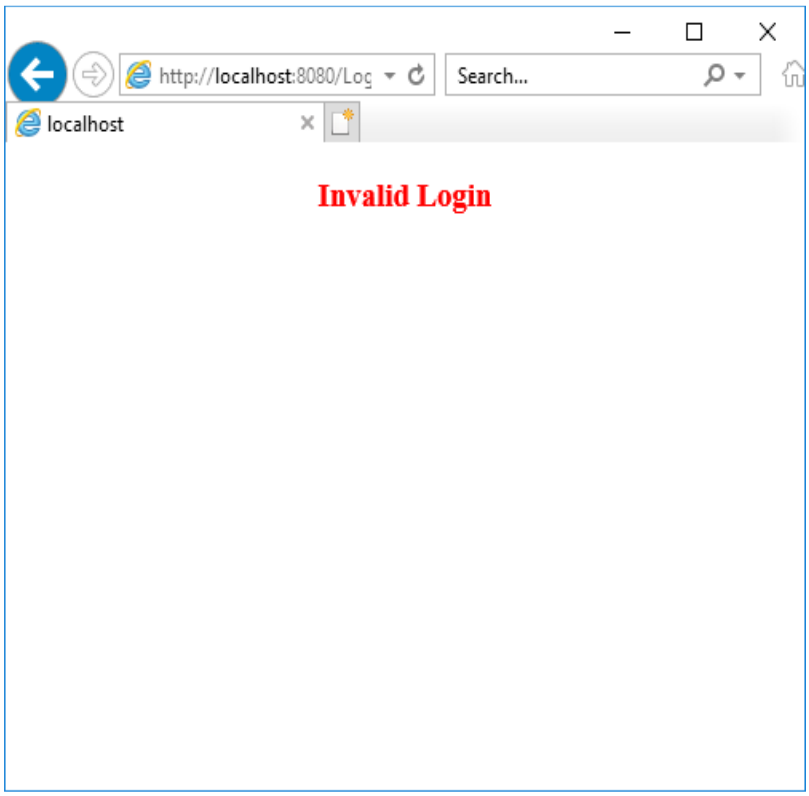
OUTPUT – HOME PAGE



SUCCESS CASE (VALID LOGIN - submission of valid user name and password)



FAILURE CASE (INVALID LOGIN - submission of invalid username / password)



NOTE

- Whenever user submits the form (pressing submit button), then next
- page will be called via form action tag
- submit button → **Event Source (Event creator)**
- action tag → **Event Handler**
- Here, the event handler (implicit calling) will be called automatically,
- whenever the submit button is pressed by the user.

Submit button

- This is a predefined event handler. Here user should write code via JSP / servlet to handle the appropriate event.

RESULT

Thus the development of JSP web application was implemented successfully.

EX.NO: 15 **SIMPLE APPLICATIONS USING PYTHON**

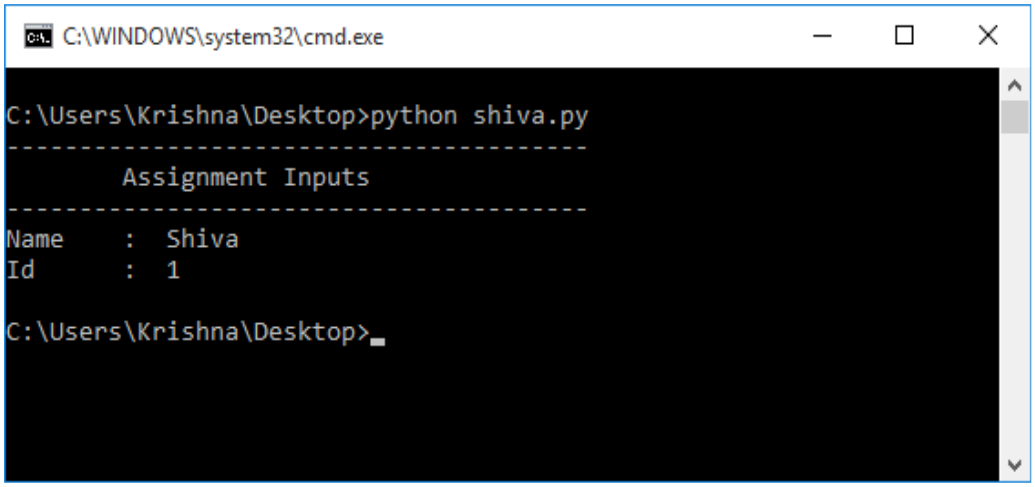
I. EXAMPLE OF ASSIGNMENT INPUTS

SOURCE CODE

(shiva.py)

```
# define variables
name="Shiva"
id=1
def disp():
print("Name \t: ",name)
print("Id \t: ",id)
print("---------------------------------------")
print("\tAssignment Inputs")
print("---------------------------------------")
# calling function using its name
disp()
```

OUTPUT



II. EXAMPLE OF COMMAND LINE INPUTS

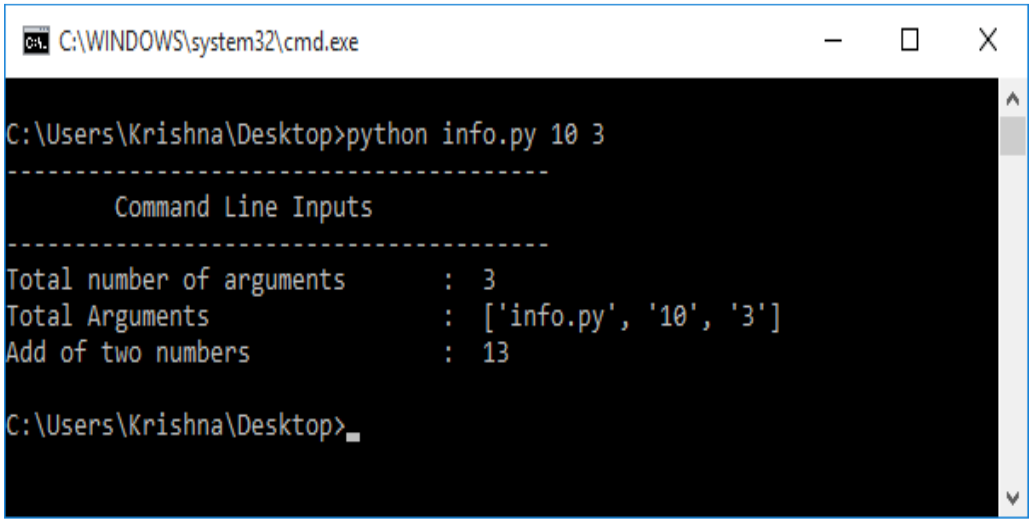
(info.py)

SOURCE CODE

```
# import sys module: to support command line arguments
import sys
print("---------------------------------------")
print("\tCommand Line Inputs")
```

```
print("---------------------------------------")
# get the total number of arguments
c=len(sys.argv)
# get 1st argument from list and convert it to integer using int() function
a=int(sys.argv[1])
# get 2nd argument from list and convert it to integer using int() function
b=int(sys.argv[2])
k=a+b
print("Total number of arguments\t: ",c)
print("Total Arguments\t\t\t: ",sys.argv)
print("Add of two numbers\t\t: ",k)
```

OUTPUT



III. EXAMPLE OF RUNTIME INPUTS

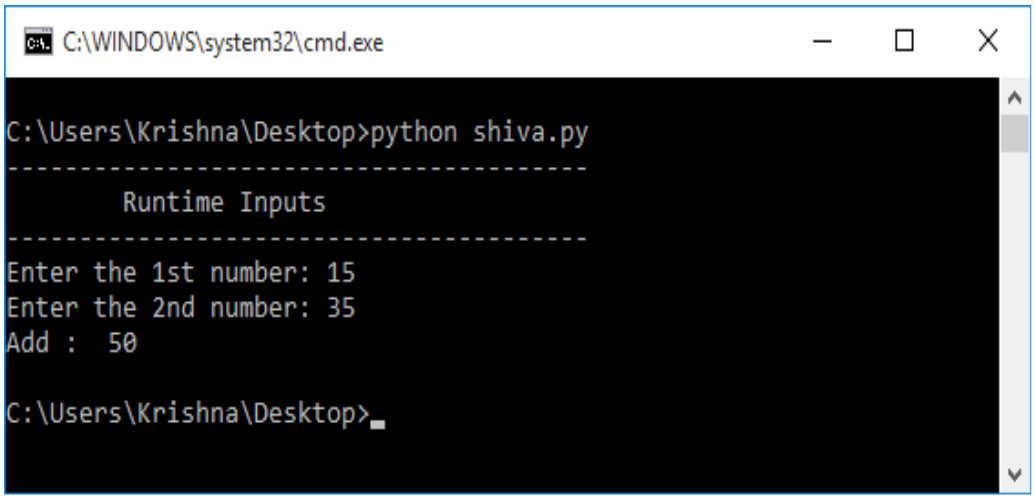
SOURCE CODE

(shiva.py)

```
# define function
def disp():
# read a number (string format)
a=input("Enter the 1st number: ")
# convert string number to int number
a=int(a)
# read another number (string format)
```

```
b=input("Enter the 2nd number: ")
# convert string number to int number
b=int(b)
c=a+b
print("Add : ",c)
print("----------------------------------------")
print("\tRuntime Inputs")
print("----------------------------------------")
# calling function using its name
disp()
```

OUTPUT



IV. EXAMPLE OF RUNTIME INPUTS

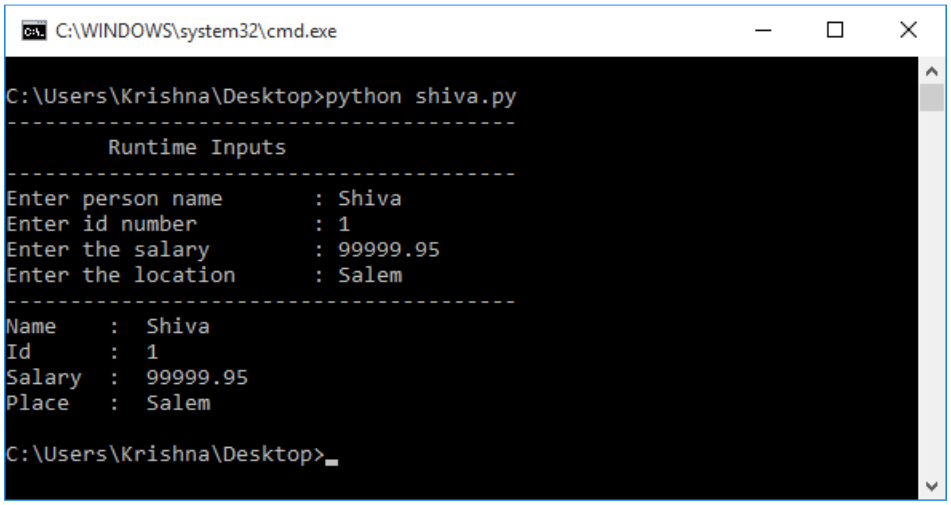
SOURCE CODE

(shiva.py)

```
# define function
def disp():
# read a name as string
name=input("Enter person name \t: ")
# read id and convert it to int
id=int(input("Enter id number \t: "))
# read salary and convert it float type
pack=float(input("Enter the salary \t: "))
```

```
# read place as string
place=input("Enter the location \t: ")
print("----------------------------------------")
# calling function
show(name,id,pack,place)
# define function
def show(name,id,pack,place):
print("Name \t: ",name)
print("Id \t: ",id)
print("Salary \t: ",pack)
print("Place \t: ",place)
print("----------------------------------------")
print("\tRuntime Inputs")
print("----------------------------------------")
# calling function using its name
disp()
```

OUTPUT



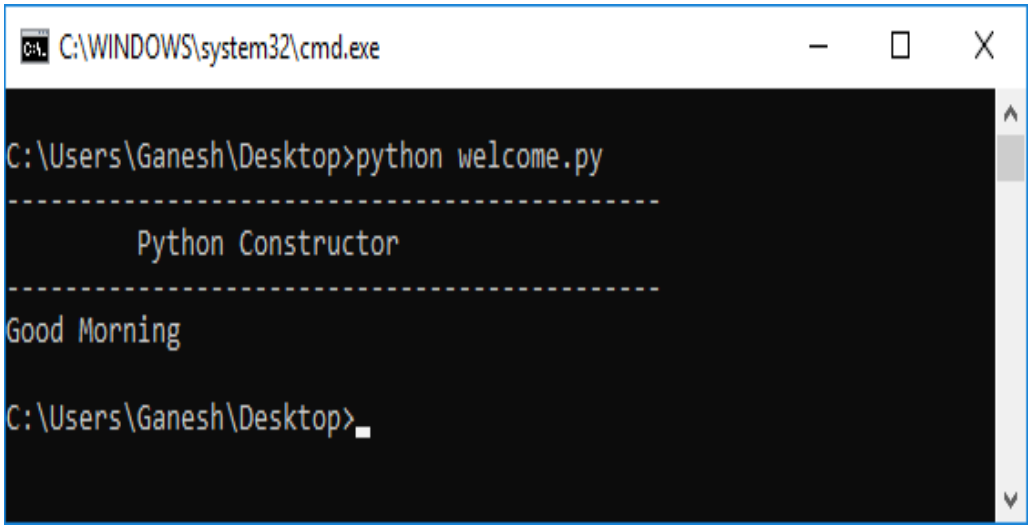
V. EXAMPLE OF DEFAULT CONSTRUCTOR (PARAMETER LESS CONSTRUCTOR)

SOURCE CODE

(welcome.py)

```
print("-----------------------------------------")
print("\t Python Constructor")
print("-----------------------------------------")
# class definition
class Hello:
def __init__(self):
print("Good Morning")
# object creation and calling constructor (default constructor)
obj=Hello()
```

OUTPUT



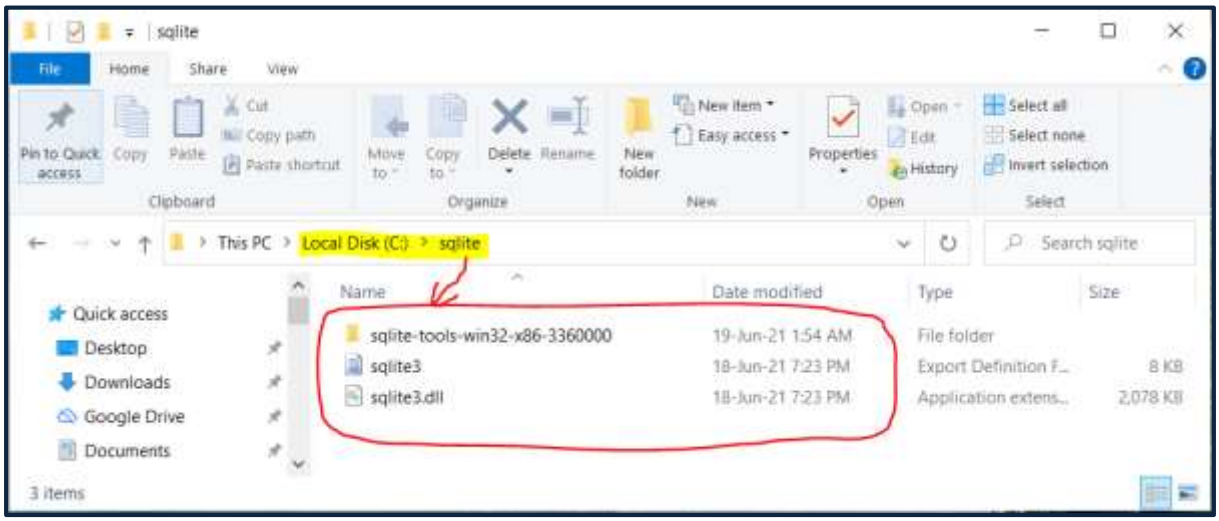
RESULT

Thus the basic programs of python applications were implemented successfully.

EX.NO: 16 DATABASE WITH PYTHON – SQLITE3

SQLITE3

- Download the following sqlite3 softwares in the SQLite home page
    1. Download sqlite-dll-win64-x64-3360000.zip (32 bit) or sqlite-dll-win64-x64-3360000.zip (64bit)
    2. Download sqlite-tools-win32-x86-3360000.zip
- Extract the zip files listed above and add the extracted contents in the folder of your computer like c/sqlite

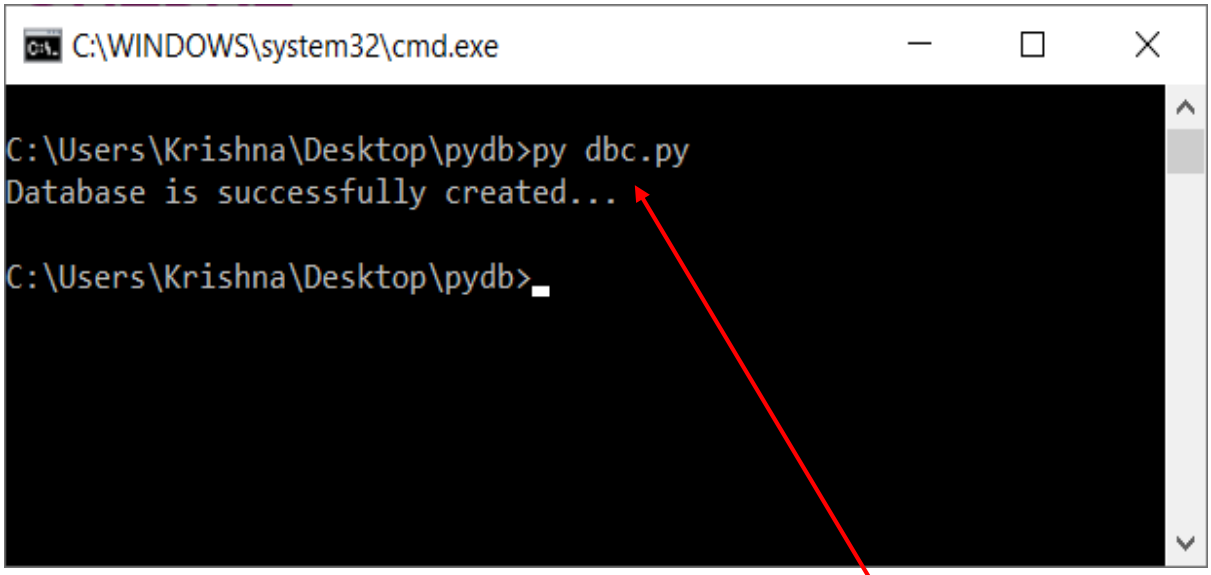


I. DATABASE CREATION

SOURCE CODE

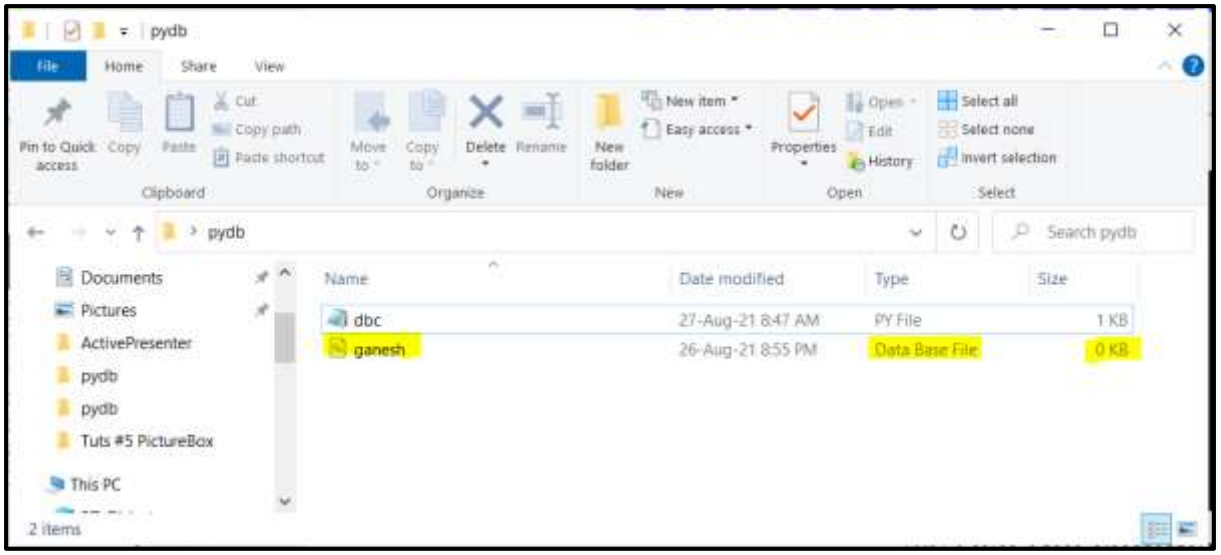
```
import sqlite3
# create the database by starting the connection with sqlite
try:
        con=sqlite3.connect("ganesh.db")
        print("Database is successfully created...")
except Error:
        print("Error in creating the database")
finally:
        con.close()
```

OUTPUT



Use the command to run the python code:
python <filename.py>
(OR)
py <filename.py>

VERIFICATION FOR NEWLY CREATED DATABASE

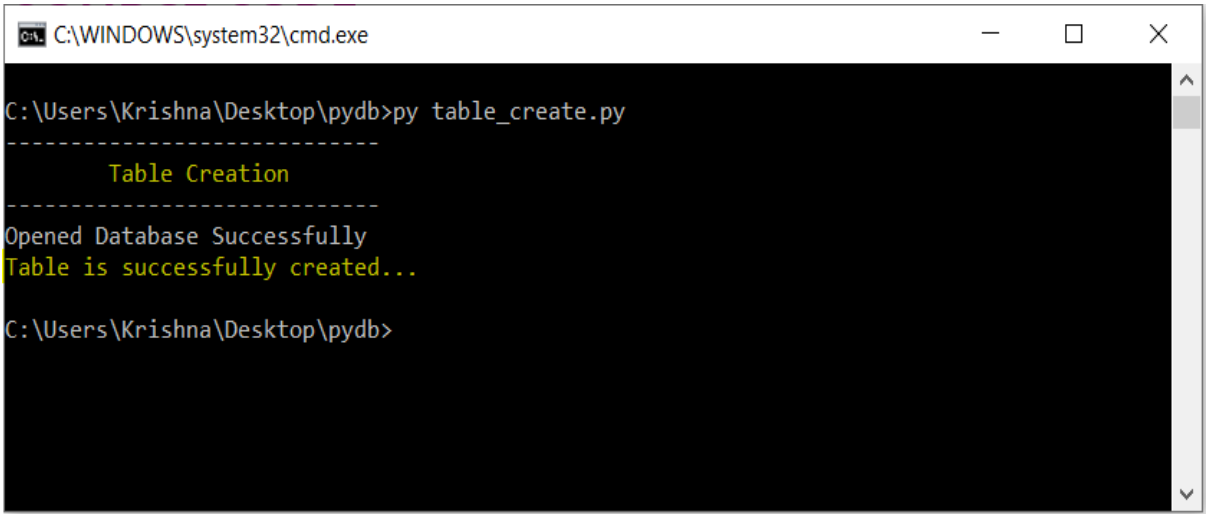


II. Table CREATION

SOURCE CODE

```
import sqlite3
print("----------------------------")
print("\tTable Creation")
print("----------------------------")
# sql create query and use if not exists for the table verification
sql="create table if not exists stud(id integer primary key, name text, dept text, cgpa
real);"
# create the database and return the connection object
con=sqlite3.connect("ganesh.db")
print("Opened Database Successfully")
# create a new table in the database using execute()
con.execute(sql)
print("Table is successfully created...")
con.close()
```

OUTPUT

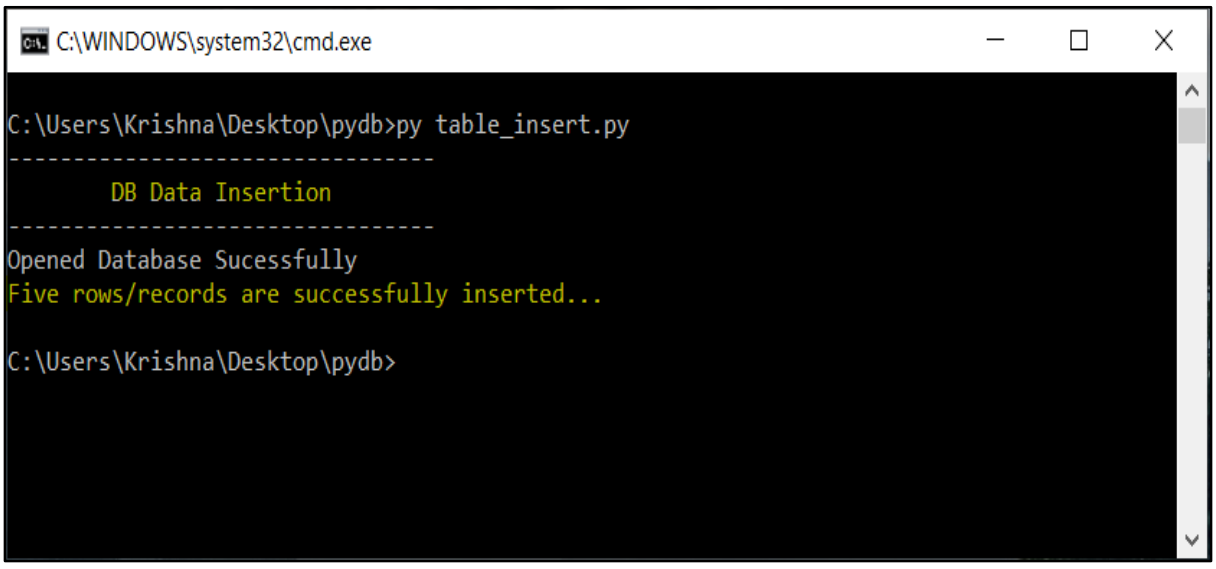


III. TABLE DATA INSERTION

SOURCE CODE

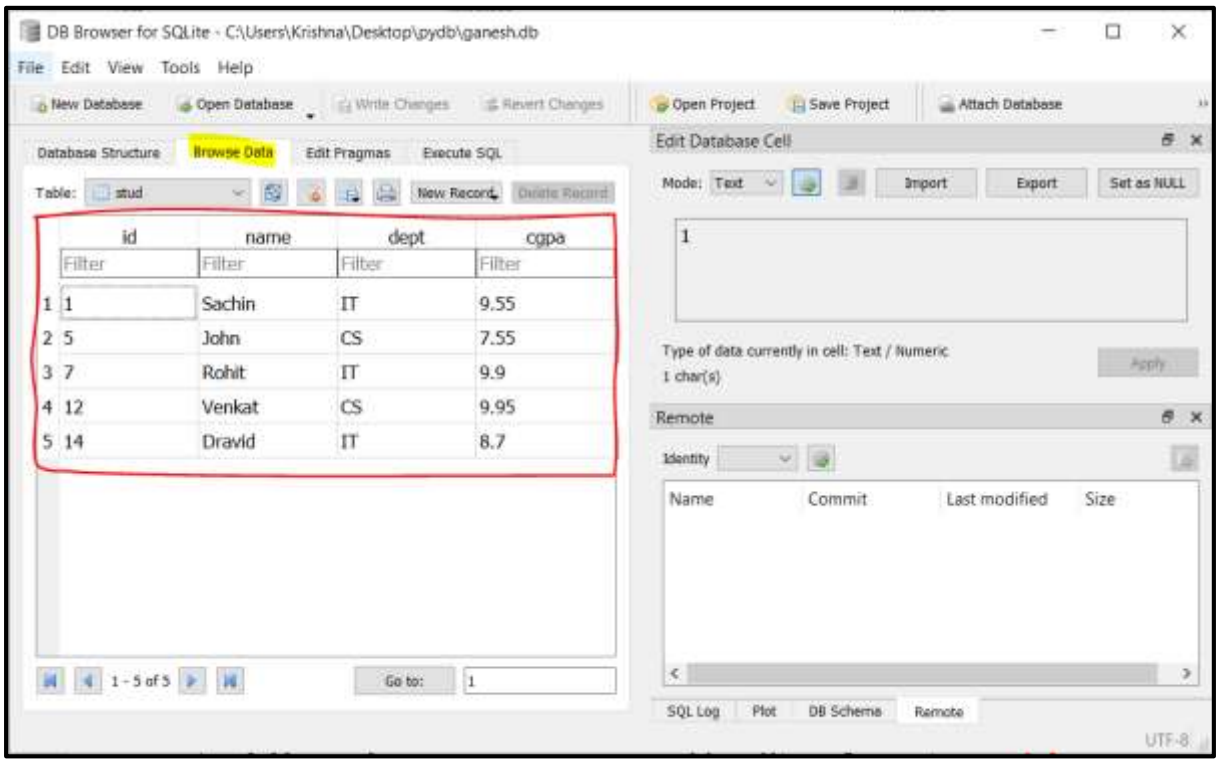
```
    print("--------------------------------")
    print("\tDB Data Insertion")
    print("--------------------------------")
# connect and open existing DB
    con=sqlite3.connect("ganesh.db")
    print("Opened Database Sucessfully")
# add the insert queries as you want in the execute() method of connection object
    con.execute("insert into stud values(1,'Sachin','IT',9.55)")
    con.execute("insert into stud values(5,'John','CS',7.55)")
    con.execute("insert into stud values(7,'Rohit','IT',9.90)")
    con.execute("insert into stud values(12,'Venkat','CS',9.95)")
    con.execute("insert into stud values(14,'Dravid','IT',8.70)")
# commit the changes
    con.commit()
    print("Five rows/records are successfully inserted...")
# close the DB
    con.close()
```

OUTPUT



PROOF FOR DATA INSERTION OPERATION

[DB Browser for SQLite – Tool]



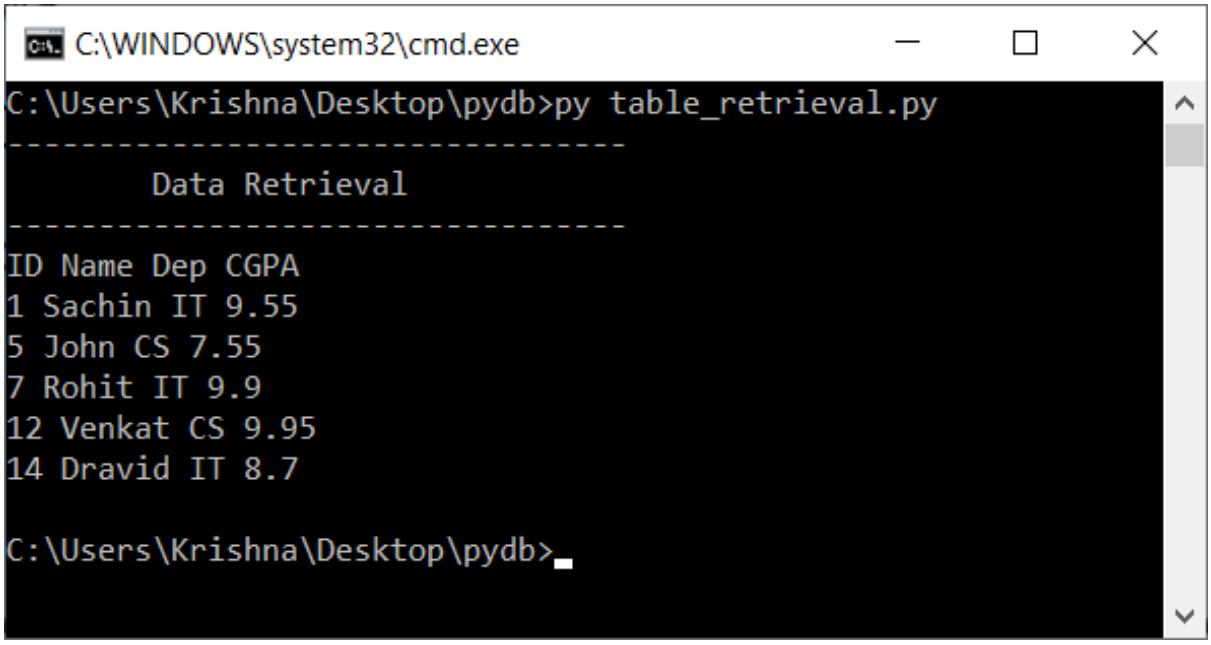
IV. DATA RETRIEVAL

SOURCE CODE

```
import sqlite3
print("---------------------------------")
print("\tData Retrieval")
print("---------------------------------")
# connect and open the existing DB
con=sqlite3.connect("ganesh.db")
# create the cursor object from connection object
cc=con.cursor()
sql="select * from stud"
# execute the sql query using cursor object
cc.execute(sql)
# read all the records from table using fetchall() method and store the result in result set variable “rs”
rs=cc.fetchall()
print("ID Name Dep CGPA")
```

```
for row in rs:
        print(row[0],row[1],row[2],row[3])
# close the DB
con.close()
```

OUTPUT



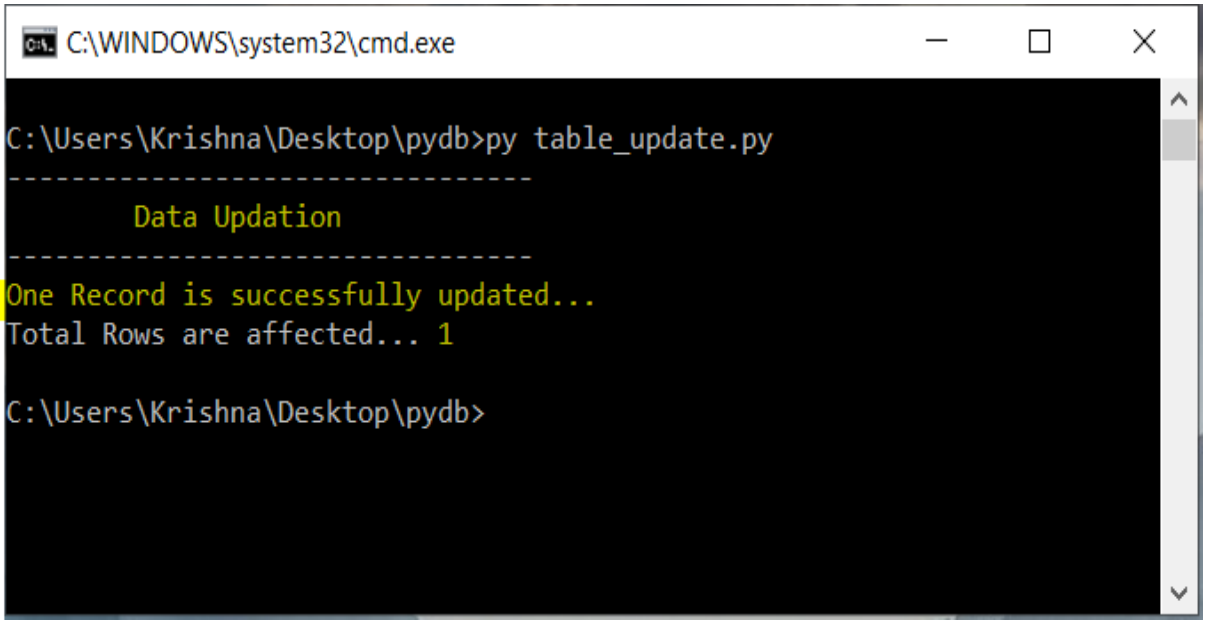
V. DATA UPDATION

SOURCE CODE

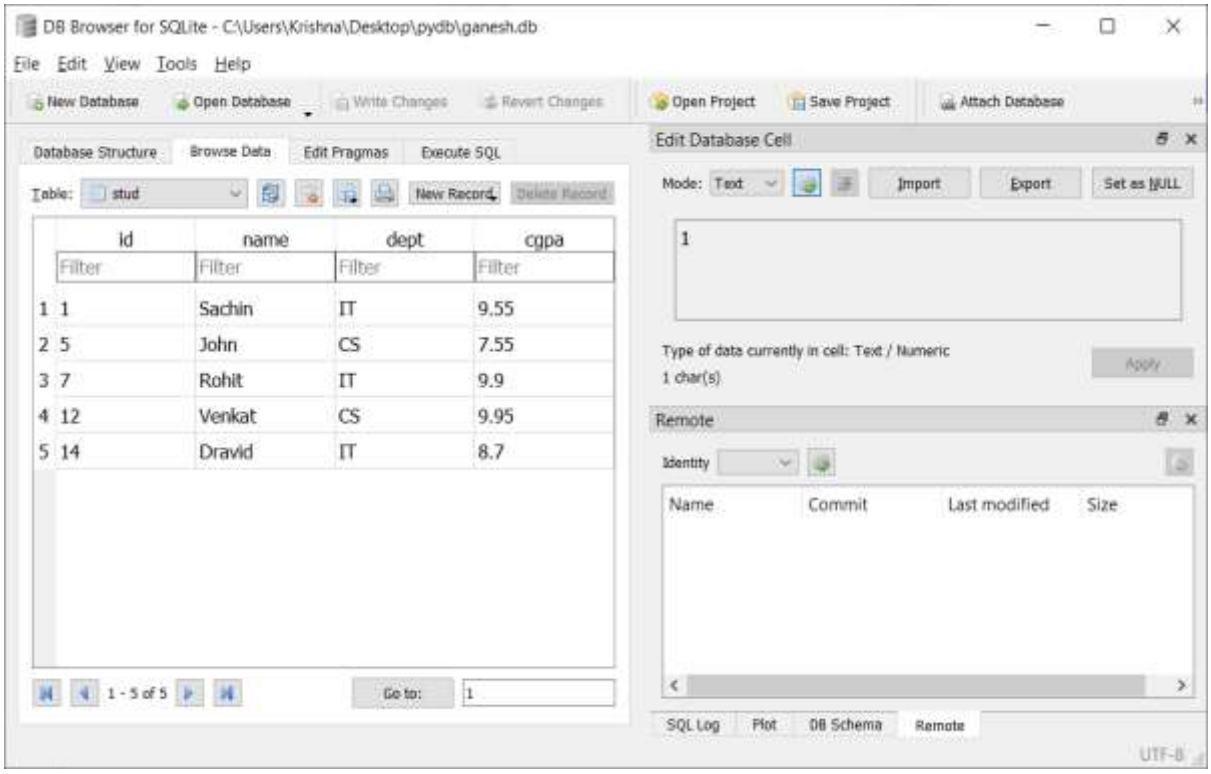
```
    import sqlite3
    print("---------------------------------")
    print("\tData Updation")
    print("---------------------------------")
# connect and open the existing DB
    con=sqlite3.connect("ganesh.db")
# define the update query
    usql="update stud set name='Velan', dept='SE' where id=5"
# execute the update query using execute() method of connection object
    con.execute(usql)
    con.commit()
print("One Record is successfully updated...")
```

```
    print("Total Rows are affected...",con.total_changes)
# close the DB
    con.close()
```

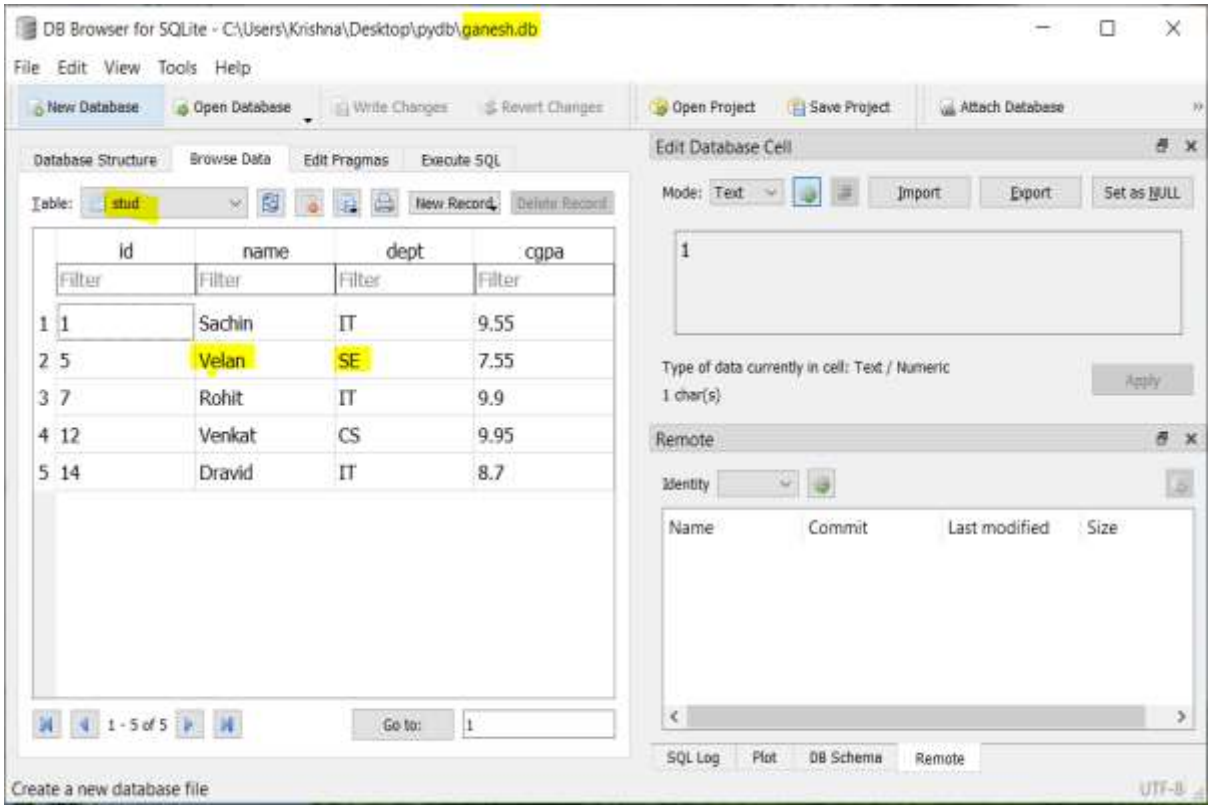
OUTPUT



BEFORE UPDATE (TABLE DATA) [DB BROWSER FOR SQLITE – TOOL]



AFTER UPDATE (TABLE DATA) [DB BROWSER FOR SQLITE – TOOL]



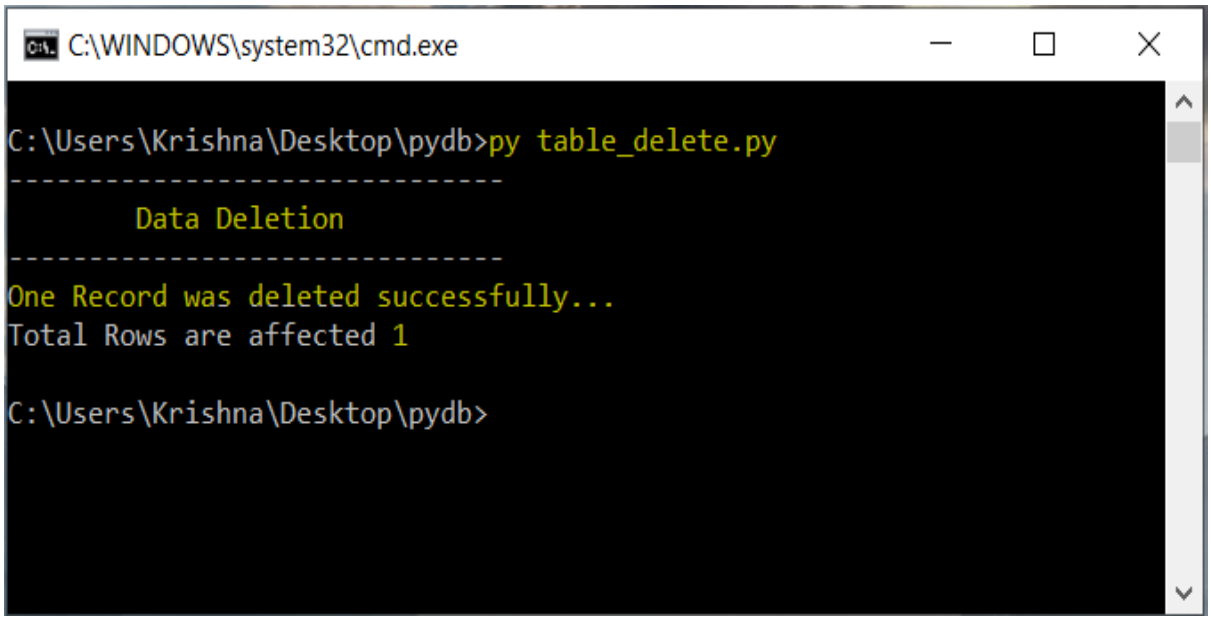
VI. DATABASE DATA DELETION

SOURCE CODE

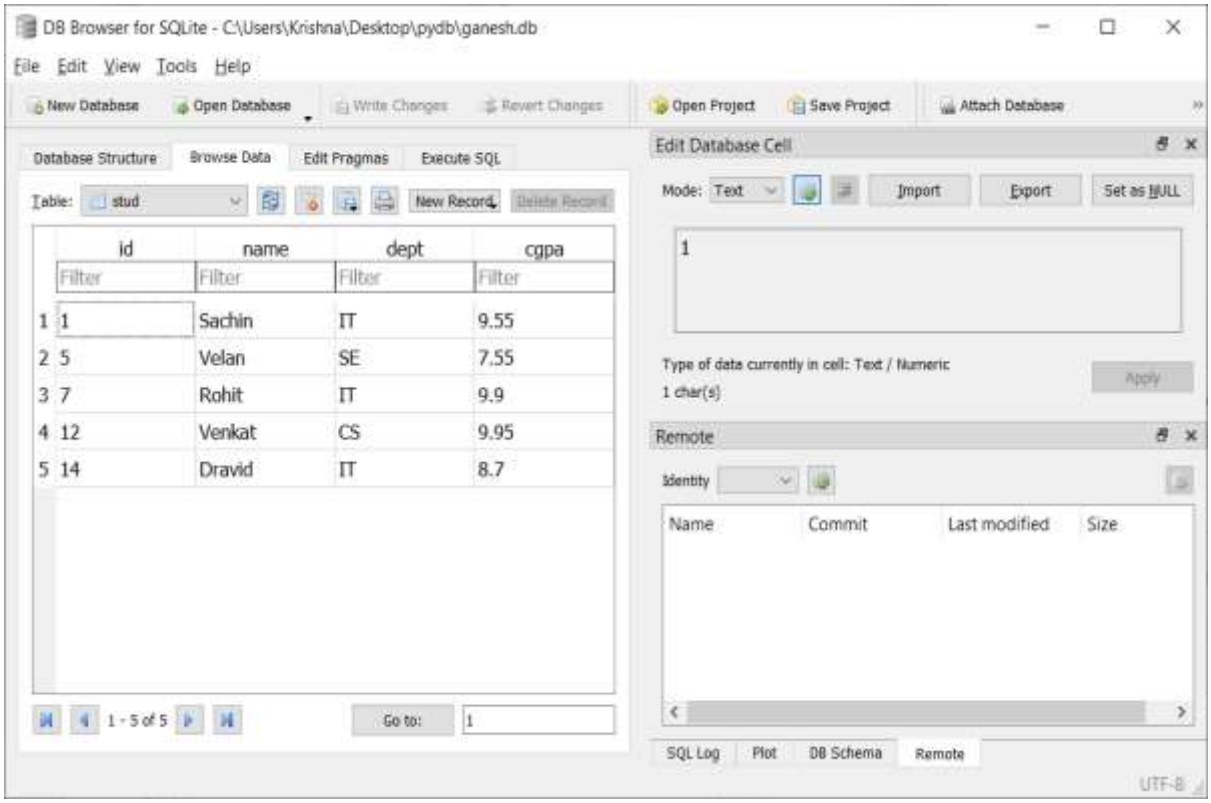
```
import sqlite3
print("------------------------------")
print("\tData Deletion")
print("------------------------------")
# connect and open the DB
con=sqlite3.connect("ganesh.db")
# delete query
dsql="delete from stud where id=7"
# execute the delete query using execute() method of connection object
con.execute(dsql)
con.commit()
print("One Record was deleted successfully...")
print("Total Rows are affected",con.total_changes)
# close the DB
```

```
con.close()
```

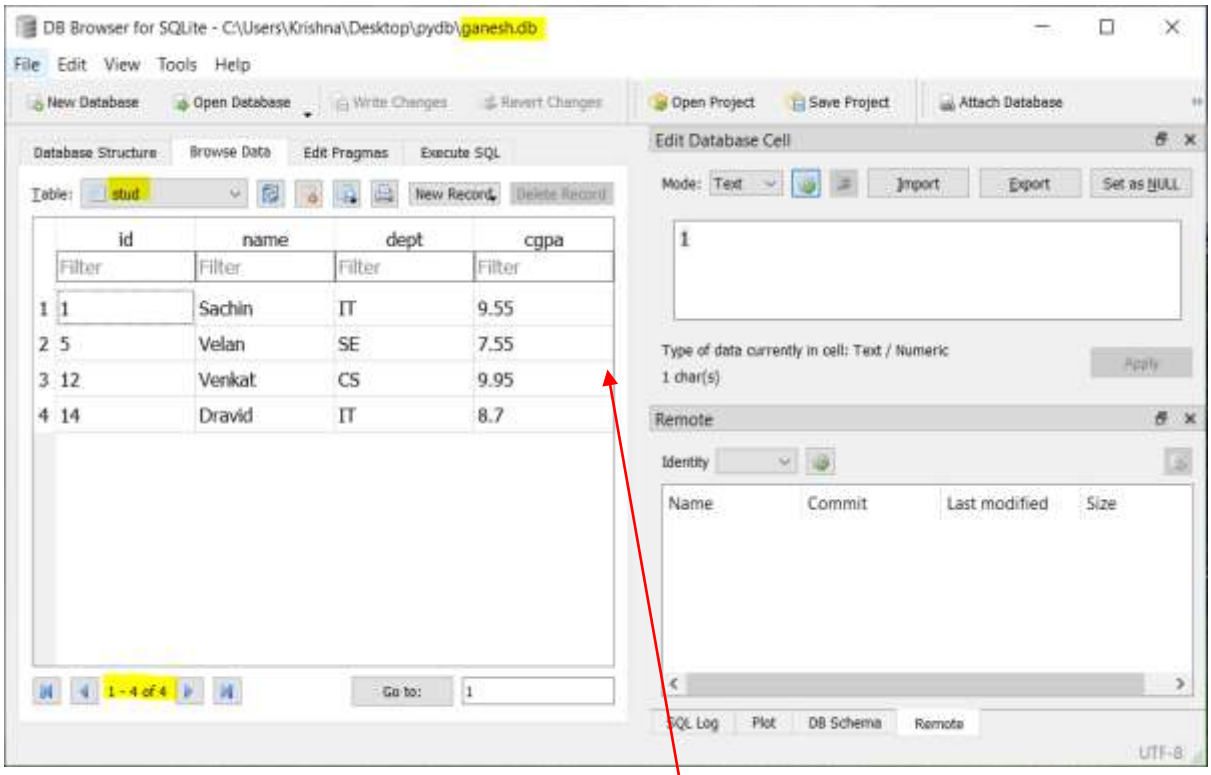
OUTPUT



BEFORE DELETE (TABLE DATA) [DB BROWSER FOR SQLITE – TOOL]



AFTER DELETE (TABLE DATA) [DB BROWSER FOR SQLITE – TOOL]



Before deletion, total records were **5**. After the deletion of one record, the total count of records are **4**.

RESULT

Thus the database sqlite3 CRUD operations are implemented using python 3.