

A SHORT-TERM INTERNSHIP REPORT

On

JAVA FULL-STACK PROGRAMMING

Submitted to Department of Bachelor of Science

By

GUNDUBOGULA HARI VEERA SIVA

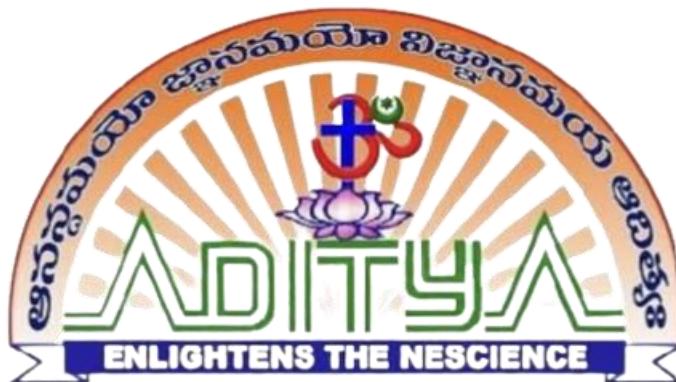
SATYANARAYANA

II BSc MPCS (SEM IV)

Under the Esteemed Guidance of

Mr. M. JANAKIRAM M.Sc(CS), M.Tech(CSE)

LECTURER OF COMPUTER SCIENCE



Department of Bachelor of Science

ADITYA DEGREE COLLEGE-KAKINADA

(Affiliated to Adikavi Nannaya University)

KAKINADA-533003, Kakinada. Dt, ANDHRA PRADESH

2021-2024



CERTIFICATE

Of Internship

Proudly Presented to :

GUNDUBOGULA HARI VEERA SIVA SATYANARAYANA

Of Aditya Degree College Has Successfully
Completed Core Java Internship During
16/09/2023 - 02/11/2023 In Adhoc Network
Tech Company

PROJECT MANAGER

ADITYA DEGREE COLLEGE

Department of Bachelor of Science



CERTIFICATE

This is to certify that The Short Term Internship entitled, "**JAVA FULL STACK PROGRAMMING**" is a bonified work of **GUNDUBOGULA HARI VEERA SIVA SATYANARAYANA**, bearing **210377102045** II BSc MPCs SEM IV submitted to the Department of Bachelor of Computer Science, Aditya Degree College, Kakinada for the academic year 2021-2024.

Internship Guide

Sri. M JANAKIRAM

M.Sc (CS),M.Tech(CSE)

Head of the Department

Sri. V.S.N. KUMAR

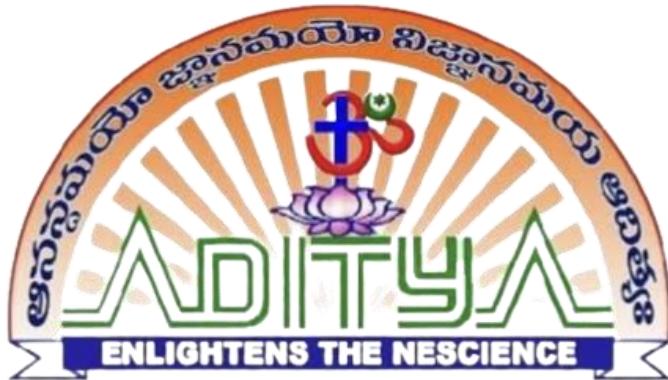
(M.Sc, M.Tech)

External Examiner

Principal

ADITYA DEGREE COLLEGE

Department of Bachelor of Science



DECLARATION BY THE STUDENT

I hereby declare that the work described in this Short Term Internship, entitled "**JAVA FULL STACK PROGRAMMING**" which is being submitted by me in partial fulfillment of the requirements for the award of degree of **Bachelor of Science** from the Department of Bachelor of Computer Science to Aditya Degree College, Kakinada under the guidance of

M. JanakiRam M.Sc(CS), M.Tech(CSE) lecturer of Computer Science in Aditya Degree College, Kakinada.

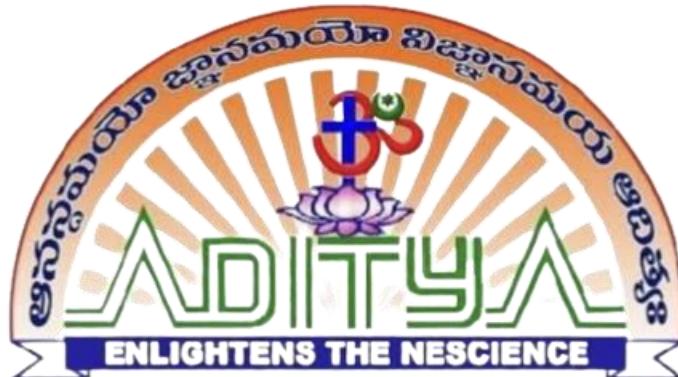
Place: Kakinada

(G.SIVA SATYANARAYANA)

Date:

ADITYA DEGREE COLLEGE

Department of Bachelor of Science



CERTIFICATE FROM THE SUPERVISOR

This is to certify that the Short Term Internship entitled, "**JAVA FULL STACK PROGRAMMING**", that is being submitted by **GUNDUBOGULA HARI VEERA SIVA SAYANARAYANA** bearing **210377102045**, II BSC MPCs (SEM IV) , which is being submitted by me in partial fulfillment of the requirements for the award of degree of **Bachelor of Science** from the Department of Bachelor of Computer Science to Aditya Degree College, bonified work carried out by him under my guidance and Supervision.

Mr. M. JanakiRam (M.Sc(CS), M.Tech(CSE))

ACKNOWLEDGEMENT

No endeavour is completed without the valuable support of others. I would like to take this opportunity to extend my sincere gratitude to all those who have contributed to the successful completion of this Short-Term Internship Project Report.

At this juncture I feel deeply honoured in expressing my sincere thanks to Miss.**DEVIKA PAKRUTHI**, CEO and Founder of Adhoc network, Visakhapatnam for making the resources available at right time and providing valuable insights leading to the successful completion of my Short-Term Internship Project Report.

It is privilege to thank **Dr.N.SESHA REDDY**, Chairman, Aditya group of institutions for providing state-of-the-Art facilities, experienced and talented faculty members.

It is privilege to thank **Dr.N.SUGUNA REDDY**, Secretary Madam, Aditya group of institutions for providing Short-Term Internship Project Report from Adhoc.

I thank **Dr. B.E.V.L.NAIDU**, Academic Director, Aditya Degree College for his continuous support and encouragement in my endeavour.

I express my deep sense of gratitude to **Mr. C. Satyanarayana Principal and N. Murthy Campus-Incharge**, for his Efforts and for giving us permission for carrying out this Short-Term Internship.

I thank **Mr. V.S.N.Kumar** , Head of the Department of Bachelor of Computer Science, Aditya Degree College-Kakinada, for supporting and encouraging me in completion of my Short-Term Internship.

Finally I thank all the faculty members of our Department who contributed their valuable suggestions in completion of Short-Term Internship report and I also put my sincere thanks to My Parents who stood with me during the whole Short-Term Internship.

- G. SIVA SATYANARAYANA

INDEX

DESCRIPTION

ABSTRACT

1. INTRODUCTION 1-4

1.1 Short Term Internship Overview	5
1.2 Existing System	5
1.3 Proposed System	6

2. SYSTEM REQUIREMENTS

2.1 Functional Requirements	8
2.2 Non-functional Requirements	9
2.3 Software Requirements Specifications	10

3. PROGRAMS

3.1 Write a Demo program to print Hello World.	11
3.2 Write a program on Data Types.	12
3.3 Write a demo program on type conversion.	13
3.4 Write a demo program on One-Dimensional array.	14
3.5 Write a demo program on Two-Dimensional array.	15-16
3.6 Write a demo program on Arithmetic operators.	17
3.7 Write a demo program on Relational operators.	18
3.8 Write a demo program on bitwise operators.	19
3.9 Write a program to print elements in matrix form	20-21
3.10 Write a program to print stars in increasing Triangular Format.	22
3.11 Write a demo program on Conditional Operator.	23

3.12 Write a program on Instance operator.	24
3.13 Write a demo program on Vehicle using Instanceof Operator.	25
3.14 Write a program on Inheritance	26
3.15 Write a demo program on If-Else statement.	27
3.16 Write a program on Nested ifelse.	28
3.17 Write a program on While Loop.	29
3.18 Write a demo program on If-else ladder.	30
3.19 Write a program on Do-While loop.	31
3.20 Write a program on Switch Demo.	32
3.21 Write a program on Pyramid.	33
3.22 Write a program on Break statement.	34
3.23 Write a program on Continue Statement.	35
3.24 Write a program to find minimum value.	36
3.25 Write a program by using this instance.	37
3.26 Write a program on usage of class and new.	38
3.27 Write a program on Default constructor.	39
3.28 Write a program on parameter constructor.	40
3.29 Write a program on constructor overloading.	41
3.30 Write a program on copy constructor.	42
3.31 Write a program on copying object without constructor.	43
3.32 Write a program on this pointer.	44
3.33 Write a program on without this pointer.	45
3.34 Write another example program on this pointer.	46
3.35 Write a program on this pointer in constructor.	47
3.36 Write a program on this keyword with method ?//this: to invoke current.	48
3.37 Write a program on this() can be used to invoked current class constructor.	49
3.38 Write a program on counter without static variable .	50
3.39 Write a program on counter with static variable.	51
3.40 Write a program to implement string methods.	52
3.41 Write a program on command line arguments.	53
3.42 Write a program to implement Hybrid Inheritance.	54-55

3.43 Write a program to implement Method Overriding with super.	56
3.44 Write a program on creating packages.	57
3.45 Write a program on interface.	58
3.46 Write a program on to implement try, catch block.	59
3.47 Write a program on to implement Exception handling mechanism of try, catch and finally blocks.	60
3.48 Write a program to create our own thread using extended Thread class.	61
3.49 Write a program to create our own Thread using Runnable Interface (Implements).	62
3.50 Write a program to implements Thread Methods using Thread class.	63-64

4. JAVA FRAMES

4.1 Java program to create frame by frame class.	66
4.2 Creating a frame by extending frame class.	67
4.3 Java program to create a button.	68
4.4 Java program to create a Border Layout.	69
4.5 Write a program to create a Flow layout.	70
4.6 Java program to create panel layout.	71
4.7 Java program to create Textfield and Label.	72-73
4.8 Java program on Checkbox.	74
4.9 Java program to create TextArea.	75
4.10 Java program to create Grid Layout.	76
4.11 Java program to create Event Layout.	77
4.12 Java program to create Focus Layout	78
4.13 Java program to create Menu Bar.	79-80
4.14 Java program to create a scroll Bar.	81
4.15 Java program to create a Key Event.	82

5. MINI PROJECT

84-145

6. CONCLUSION

146

ABSTRACT

This Short Term Internship revolves around the comprehensive development of a Java-based full-stack web application, aiming to deliver a robust and user-friendly platform.

The internship also delved into object-oriented programming paradigms, covering various aspects of constructors and demonstrating the use of this pointer to enhance code readability and maintainability. The implementation of counters and string methods further enriched the understanding of object-oriented concepts.

Advanced topics such as Hybrid Inheritance, Method Overriding with super, and the creation of packages were explored to promote code modularity and reusability. The internship culminated in the study of multi-threading and exception handling mechanisms, showcasing practical implementations of these crucial Java features.

In summary, this Java internship equipped me with a diverse skill set ranging from basic programming constructs to advanced object-oriented principles. The projects undertaken not only improved my proficiency in Java but also instilled a problem-solving mindset essential for real-world application development.

INTRODUCTION

Introduction

Launched on 16th January, 2016, the Startup India Initiative has rolled out several programs with the objective of supporting entrepreneurs, building a robust startup ecosystem and transforming India into a country of job creators instead of job seekers. These programs are managed by a dedicated Startup India Team, which reports to the Department for Industrial Policy and Promotion (DPIIT). This startup culture, or startup mindset, features several key characteristics, including a commitment to innovation and a willingness to take risks and make decisions quickly. This atmosphere attracts potential workers who seek out that environment.

A startup is in the first stage of operations and comprises one or more entrepreneurs. The primary aim is to answer market demand by creating new and innovative products or services.

Startups begin with high costs and have limited revenue. Also, they do not have a developed business model and lacks adequate capital to move to the next phase. As a result, these companies seek funding from various sources, such as venture capitalists, angel investors and banks. Investors or lenders might offer additional funds for a share of future profits and partial ownership. Often, these companies use seed capital for investing in research and developing business plans. Research helps them determine the demand for a specific product and a business plan outlines the company's goals and marketing strategies.

About Adhoc Network:-

Adhoc Network is started in the year 2020 at Visakhapatnam by an young Women entrepreneur Miss. Devika Pakruthi Founder & CEO with an intention to provide employment opportunities to the youth and also to impart the best quality training and practical exposure to the Students which enhances their employability Skills. Her journey started and collapsed with the wide spread of Covid-19 but her determination and aspirations

made her journey more futuristic and she never gave up the thought to GIVE-UP. This is where Devika made her dreams come true and alive. She never expected with a sole objective of making Profit but her determination to impart the quality training made her to reach the peaks of success at the young age.

Adhoc Network-we are proud to have been awarded a Hat-trick of Awards. This Achievement is a testament to our commitment to Excellence and Innovation in the Software development Best heading company in the market. Devika Pakruthi, a name synonymous with innovation, empowerment, and success. As the proud recipient of the Young Entrepreneur Award, Best Women-Led Startup Award, Women Rising Star of the Year Award and Youngest CEO of the Year Award, Devika Pakruthi has etched her name into the annals of contemporary business history.

Vision, Mission and values of the Organization:-

VISION:-

Due vision is to be a leading global provider of interactive and reliable Software Solution empowering business to Thrive in the digital age.

MISSION:-

Our mission is to develop cutting-edge Software Solutions that Solve compare business challenges, enhance Operations efficiency and drive Sustainable growth for our client we strive to deliver exceptional value by leveraging emerging technologies, fostering Strategic partnerships and maintaining a Customer, Centric approach Values

- Innovation
- Excellence
- Collaboration
- Integrity

- Customer Centricity
- Continuous Learning

We are providing services to the following Clients-

- MTA Canada
- Aleph Nation-Wide Solutions India
- Botta Associates

Introduction to Java:-

Java is an Object-Oriented, General-Purpose Programming Language. It is a widely used robust technology. It was developed by James Gosling by sun microsystems. Later, it was named as Oak and JDK (Java Development Kit) developed in 1950s. Java programs can run on any device with a Java Virtual Machine (JVM), ensuring platform independence i.e, it runs on any platforms like Windows, Linux etc. Java syntax is similar to C++, making it accessible to programmers familiar with C-style languages.

Java is an object oriented because everything in a java program is an object. It includes classes and objects. Java has a better security to protect the data by setting passwords and easy to handle threats and errors.

Inheritance, polymorphism, and encapsulation are the three main OOP features supported in Java.

1.1. SHORT TERM INTERSHIP

OVERVIEW

This Java full-stack web application project is designed to create a responsive and efficient platform, combining front-end and back-end development. The front-end utilizes Angular, React, or Vue.js for an intuitive user interface with a focus on responsiveness across various devices. On the back-end, Spring Boot is employed to establish a scalable server-side architecture, incorporating a relational database for secure data management. Rigorous testing, including unit, integration, and end-to-end testing, is implemented to ensure application reliability, with automation through continuous integration and deployment pipelines. Security measures, such as encryption and authentication protocols, are integrated to safeguard sensitive information. Overall, this project adopts modern development practices to deliver a high-performance web application that prioritizes simplicity, efficiency, and a positive user experience.

1.2. EXISTING SYSTEM :

The current system in place lacks the necessary efficiency and responsiveness required for modern web applications. It exhibits limitations in terms of user interface design, often resulting in a suboptimal user experience. Additionally, the absence of a robust back-end architecture hinders scalability and secure data

management. Furthermore, the absence of comprehensive testing practices increases the risk of reliability issues, and security measures are minimal, posing a potential threat to sensitive information. In summary, the existing system falls short in meeting contemporary standards for web application development, necessitating the implementation of a more sophisticated and streamlined solution.

1.3. PROPOSED SYSTEM :

To ensure reliability, the proposed system will implement rigorous testing methodologies, including unit testing, integration testing, and end-to-end testing. Continuous integration and deployment pipelines will be established to automate the testing and deployment processes, reducing the risk of errors and enhancing development efficiency.

Security measures, including encryption and authentication protocols, will be implemented to safeguard sensitive information, addressing the shortcomings of the existing system. The proposed system thus aims to deliver a high-performance web application that adheres to modern development practices, ensuring simplicity, efficiency, and an improved user experience.

APPLICATIONS OF PROPOSED

SYSTEM :-

- Intuitive user interface enhances the learning experience for students and educators.
- Robust security measures protect confidential HR information.
- Intuitive design improves collaboration and task management.
- Reliable back-end architecture ensures real-time data synchronization.

SYSTEM REQUIREMENTS

1. Functional Requirements:-

User Authentication: Users should be able to register, log in, and log out securely.

User Interface: The system should provide an intuitive and responsive user interface, supporting multiple devices and screen sizes.

Data Management: The application should enable users to perform CRUD (Create, Read, Update, Delete) operations on relevant data.

Search Functionality: Users should be able to search and filter data efficiently based on specified criteria.

Security Measures: The system should implement encryption and authentication protocols to ensure the confidentiality and integrity of user data.

2. Non-functional Requirements:-

Performance: The system should respond to user requests within acceptable time frames, even under peak loads.

Scalability: The architecture should be scalable to accommodate a growing user base and increasing data volumes.

Compatibility: The system should be compatible with various web browsers and operating systems.

3. Software Requirements

Specifications (SRS):-

Operating System: The application should be compatible with Windows, Linux, and macOS.

Web Server: Apache or Nginx for hosting the web application.

Database Management System (DBMS): MySQL or PostgreSQL for efficient data storage and retrieval.

Programming Language: Java for back-end development.

4. Hardware Components:-

Server: A dedicated server or cloud-based infrastructure to host the web application.

Processor: Multi-core processors for handling concurrent user requests.

Memory (RAM): A sufficient amount of RAM to support the application's performance requirements.

5. Network Components:-

Internet Connection: A high-speed and reliable internet connection for seamless user interactions.

Firewall: Network firewall to protect against unauthorized access and potential security threats.

Load Balancer: If applicable, a load balancer to distribute incoming traffic across multiple servers for optimal performance and scalability.

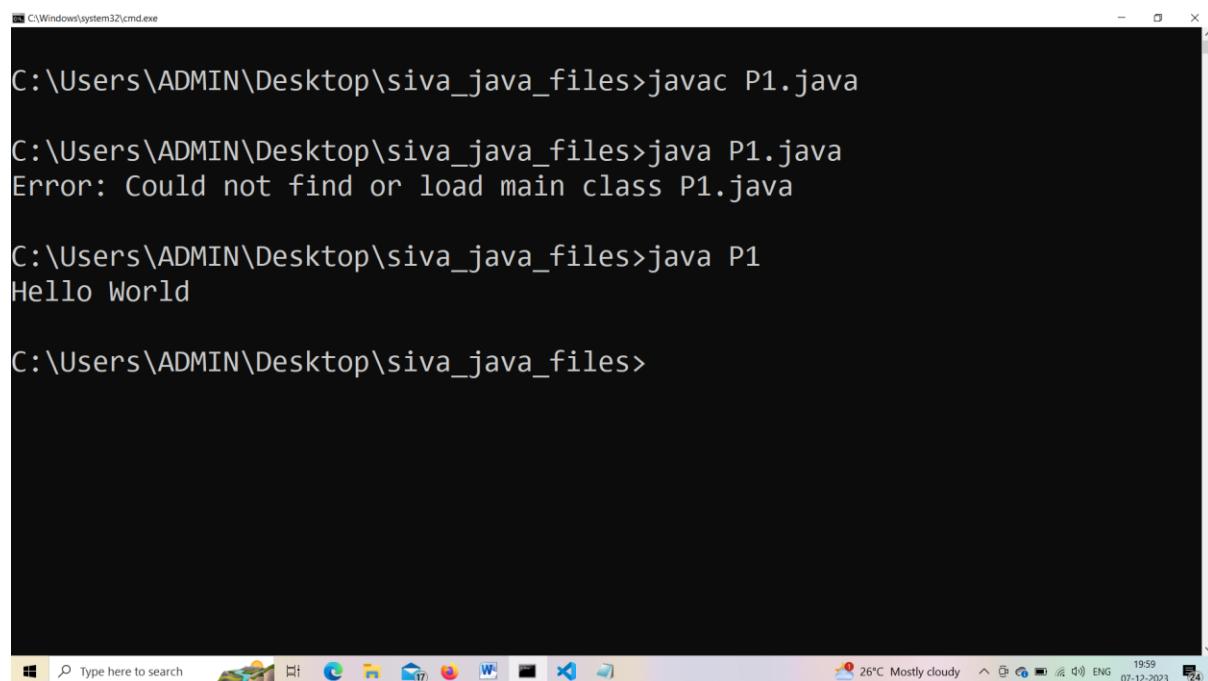
PROGRAMS

1. Write a program to display Hello World.

Program:

```
class P1
{
    public static void main(String args[])
    {
        System.out.println("Hello World");
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command line shows the following sequence of operations:

- The user types 'javac P1.java' and presses Enter. The output is: C:\Users\ADMIN\Desktop\siva_java_files>javac P1.java
- The user types 'java P1.java' and presses Enter. The output is: C:\Users\ADMIN\Desktop\siva_java_files>java P1.java Error: Could not find or load main class P1.java
- The user types 'java P1' and presses Enter. The output is: C:\Users\ADMIN\Desktop\siva_java_files>java P1 Hello World
- The user types 'javac P1.java' again and presses Enter. The output is: C:\Users\ADMIN\Desktop\siva_java_files>

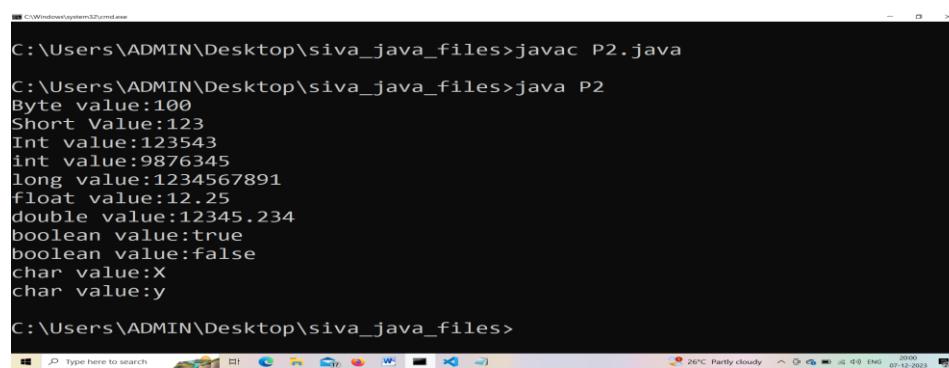
The taskbar at the bottom of the screen shows various pinned icons and the system tray with weather information (26°C, Mostly cloudy), date (07-12-2023), and time (19:59).

2. Write a program for datatypes

Program:

```
public class P2
{
    public static void main(String args[])
    {
        byte b=100;
        short s=123;
        int v=123543;
        int calc=9876345;
        long amountVal=1234567891;
        float interestRate=12.25f;
        double sineVal=12345.234d;
        boolean flag=true;
        boolean val=false;
        char ch1=88;
        char ch2='y';
        System.out.println("Byte value:"+b);
        System.out.println("Short Value:"+s);
        System.out.println("Int value:"+v);
        System.out.println("int value:"+calc);
        System.out.println("long value:"+amountVal);
        System.out.println("float value:"+interestRate);
        System.out.println("double value:"+sineVal);
        System.out.println("boolean value:"+flag);
        System.out.println("boolean value:"+val);
        System.out.println("char value:"+ch1);
        System.out.println("char value:"+ch2);
    }
}
```

Output:-



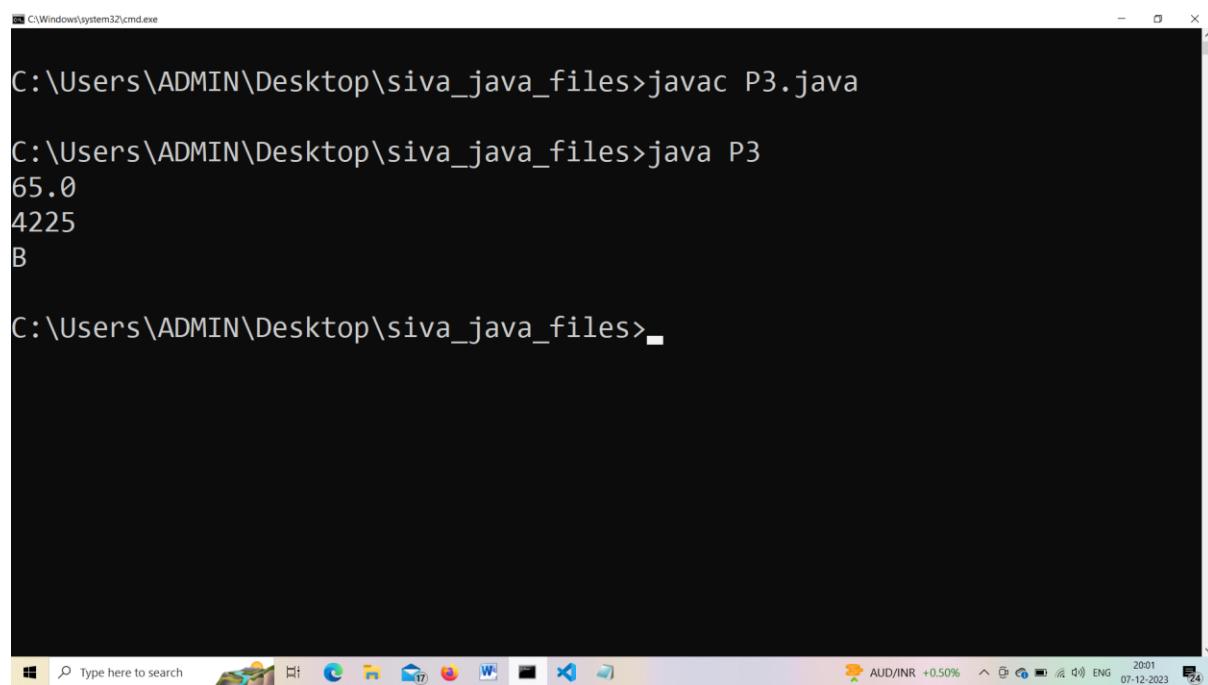
```
C:\Users\ADMIN\Desktop\siva_java_files>javac P2.java
C:\Users\ADMIN\Desktop\siva_java_files>java P2
Byte value:100
Short Value:123
Int value:123543
int value:9876345
long value:1234567891
float value:12.25
double value:12345.234
boolean value:true
boolean value:false
char value:X
char value:y
```

3. Write a program for type conversion.

Program:

```
class P3
{
    public static void main(String args[])
    {
        char ch1='A';
        double d1=ch1;
        System.out.println(d1);
        System.out.println(ch1*ch1);
        double d2=66.0;
        char ch2=(char)d2;
        System.out.println(ch2);
    }
}
```

Output:-



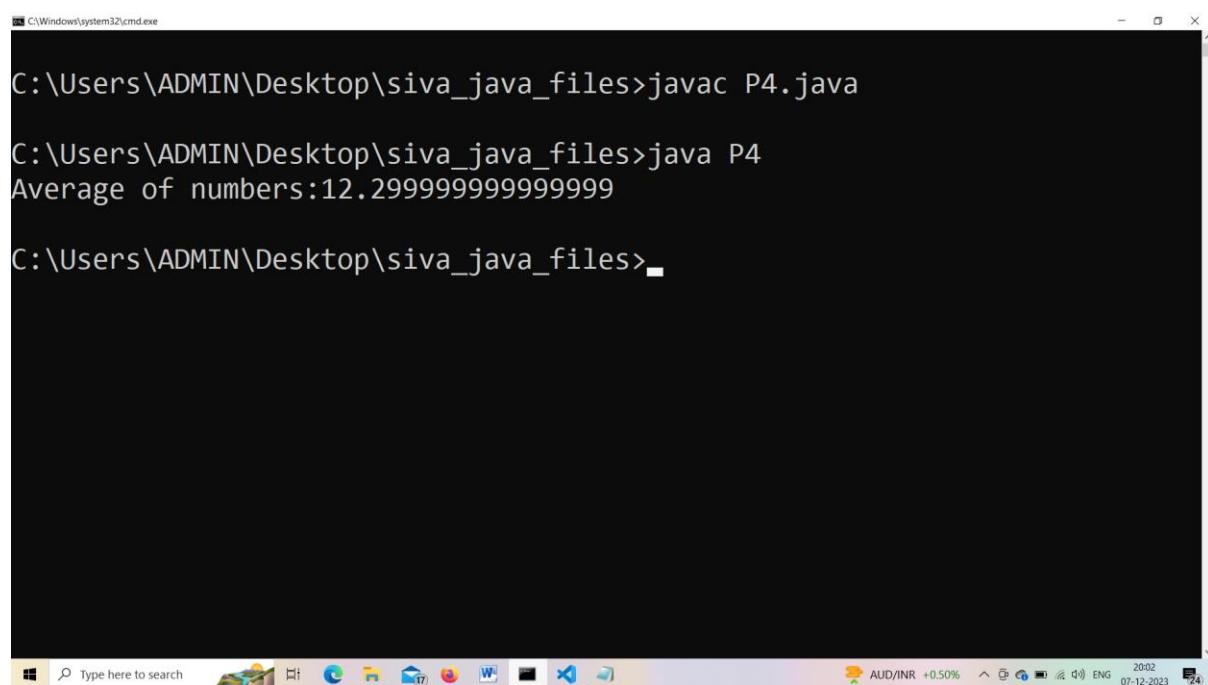
```
C:\Users\ADMIN\Desktop\siva_java_files>javac P3.java
C:\Users\ADMIN\Desktop\siva_java_files>java P3
65.0
4225
B
```

4. Write a program to know the average of numbers.

Program:

```
class P4
{
    public static void main(String args[])
    {
        double num[]={10.1,11.2,12.3,13.4,14.5};
        double result=0;
        for(int i=0;i<5;i++)
            result=result+num[i];
        System.out.println("Average of numbers:"+result/5);
    }
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac P4.java
C:\Users\ADMIN\Desktop\siva_java_files>java P4
Average of numbers:12.29999999999999
C:\Users\ADMIN\Desktop\siva_java_files>
```

5. Write a demo program on Two-Dimensional Array.

Program:

```
class TwoDim
{
    public static void main(String args[])
    {
        int a[][]=new int[2][2];
        int b[][]=new int[2][2];
        a[0][0]=1;
        a[0][1]=2;
        a[1][0]=3;
        a[1][1]=4;
        b[0][0]=5;
        b[0][1]=6;
        b[1][0]=7;
        b[1][1]=8;
        int c[][]=new int[2][2];
        System.out.println("Elements of first matrix:");
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                System.out.println(a[i][j]);
            }
        }
        System.out.println("Elements of second matrix:");
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                System.out.println(b[i][j]);
            }
        }
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                c[i][j]=a[i][j]+b[i][j];
            }
        }
    }
}
```

```
}
```

```
System.out.println("Addition of two matrices:");
```

```
for(int i=0;i<2;i++)
```

```
{
```

```
for(int j=0;j<2;j++)
```

```
{
```

```
System.out.println(c[i][j]);
```

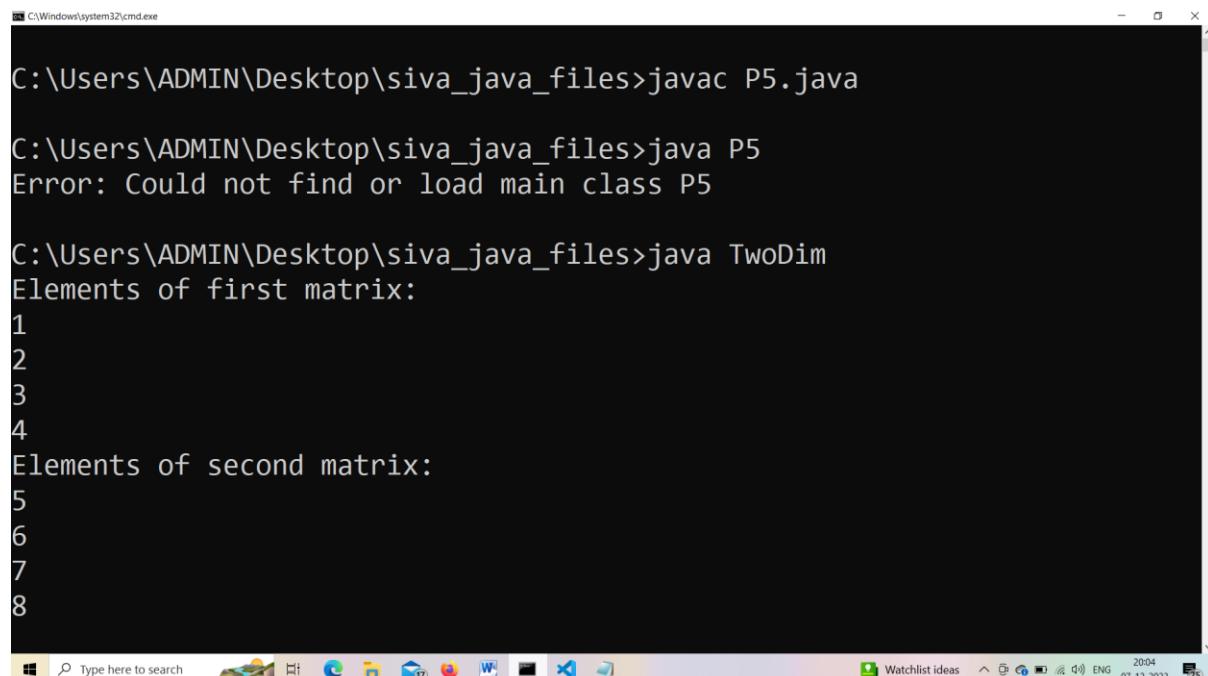
```
}
```

```
}
```

```
}
```

```
}
```

Output:-



C:\Users\ADMIN\Desktop\siva_java_files>javac P5.java

C:\Users\ADMIN\Desktop\siva_java_files>java P5

Error: Could not find or load main class P5

C:\Users\ADMIN\Desktop\siva_java_files>java TwoDim

Elements of first matrix:

1
2
3
4

Elements of second matrix:

5
6
7
8

The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system22\cmd.exe'. The user has run 'javac P5.java' which compiles the file successfully. However, when they try to run 'java P5', it fails with an error message stating 'Error: Could not find or load main class P5'. This is because the class name 'P5' does not match the file name 'P5.java'. The user then runs 'java TwoDim' which executes correctly, displaying the elements of two 2x4 matrices. The desktop background is visible at the top, and the taskbar at the bottom shows icons for File Explorer, Edge, Mail, and File Explorer again.

6. Write a demo program on Arithmetic Operations.

Program:-

```
import java.util.Scanner;
class BasicArithmeticDemo
{
    public static void main(String args[])
    {
        int a,b;
        System.out.println("Enter value of a and b:");
        Scanner input=new Scanner(System.in);
        a=input.nextInt();
        b=input.nextInt();
        System.out.println("Addition:"+ (a+b));
        System.out.println("Subtraction:"+ (a-b));
        System.out.println("Multiplication:"+ (a*b));
        System.out.println("Division:"+ (a/b));
        System.out.println("Percentage:"+ (b%a));
    }
}
```

Output:-

```
C:\Users\ADMIN\Desktop\siva_java_files>javac P6.java
C:\Users\ADMIN\Desktop\siva_java_files>java BasicArithmeticDemo
Enter value of a and b:
25
21
Addition:46
Subtraction:4
Multiplication:525
Division:1
Percentage:21

C:\Users\ADMIN\Desktop\siva_java_files>
```

7. Write a demo program on Relational Operators.

Program:-

```
import java.util.Scanner;
class P7
{
    public static void main(String args[])
    {
        int a,b;
        System.out.println("Enter value of a and b:");
        Scanner input=new Scanner(System.in);
        a=input.nextInt();
        b=input.nextInt();
        System.out.println("Greater than operator(a>b):"+(a>b));
        System.out.println("Greater than or equal to operator(a>=b):"+(a>=b));
        System.out.println("Less than operator(a<b):"+(a<b));
        System.out.println("Lesser than or equal to operator(a<=b):"+(a<=b));
        System.out.println("Not equal to operator(a!=b):"+(a!=b));
        System.out.println("Greater than operator(a==b):"+(a==b));
    }
}
```

Output:-

```
C:\Users\ADMIN\Desktop\siva_java_files>javac P7.java
C:\Users\ADMIN\Desktop\siva_java_files>java P7
Enter value of a and b:
23
45
Greater than operator(a>b):false
Greater than or equal to operator(a>=b):false
Less than operator(a<b):true
Lesser than or equal to operator(a<=b):true
Not equal to operator(a!=b):true
Greater than operator(a==b):false

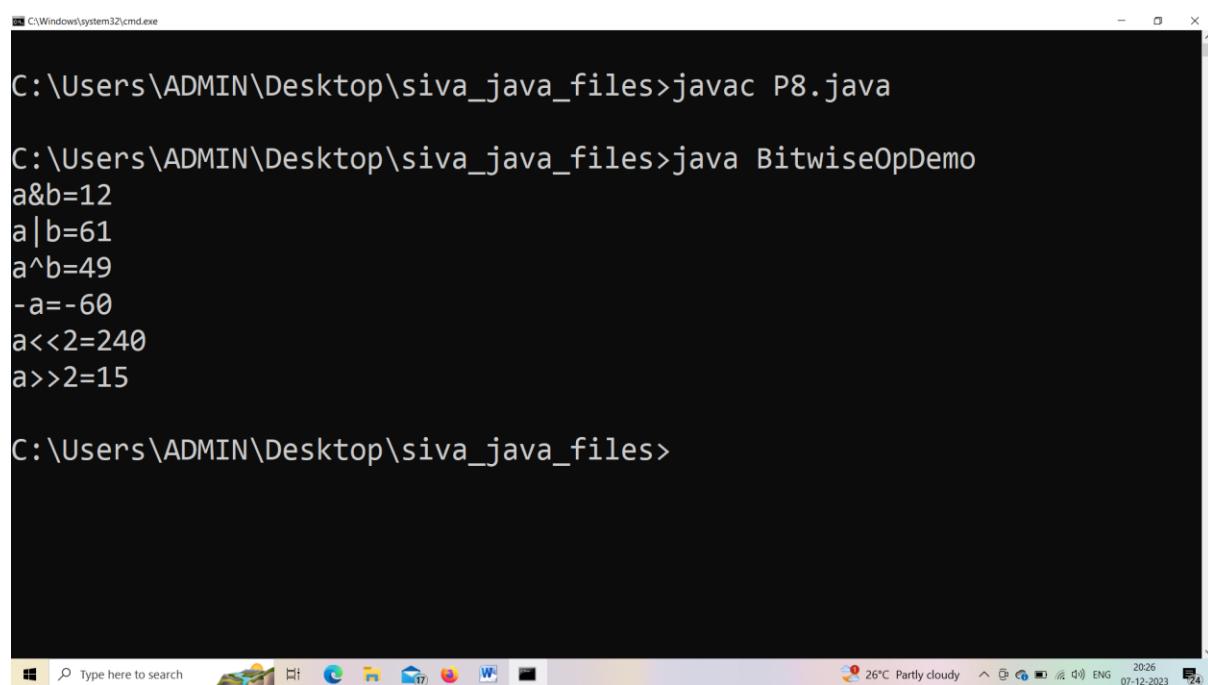
C:\Users\ADMIN\Desktop\siva_java_files>
```

8. Write a demo program on Bitwise Operators

Program:

```
import java.util.Scanner;
class BitwiseOpDemo{
public static void main(String args[]){
int a=60;
int b=13;
int c=0;
c=a&b;
System.out.println("a&b="+c);
c=a|b;
System.out.println("a|b="+c);
c=a^b;
System.out.println("a^b="+c);
c=-a;
System.out.println("-a="+c);
c=a<<2;
System.out.println("a<<2="+c);
c=a>>2;
System.out.println("a>>2="+c);
}
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac P8.java
C:\Users\ADMIN\Desktop\siva_java_files>java BitwiseOpDemo
a&b=12
a |b=61
a^b=49
-a=-60
a<<2=240
a>>2=15
```

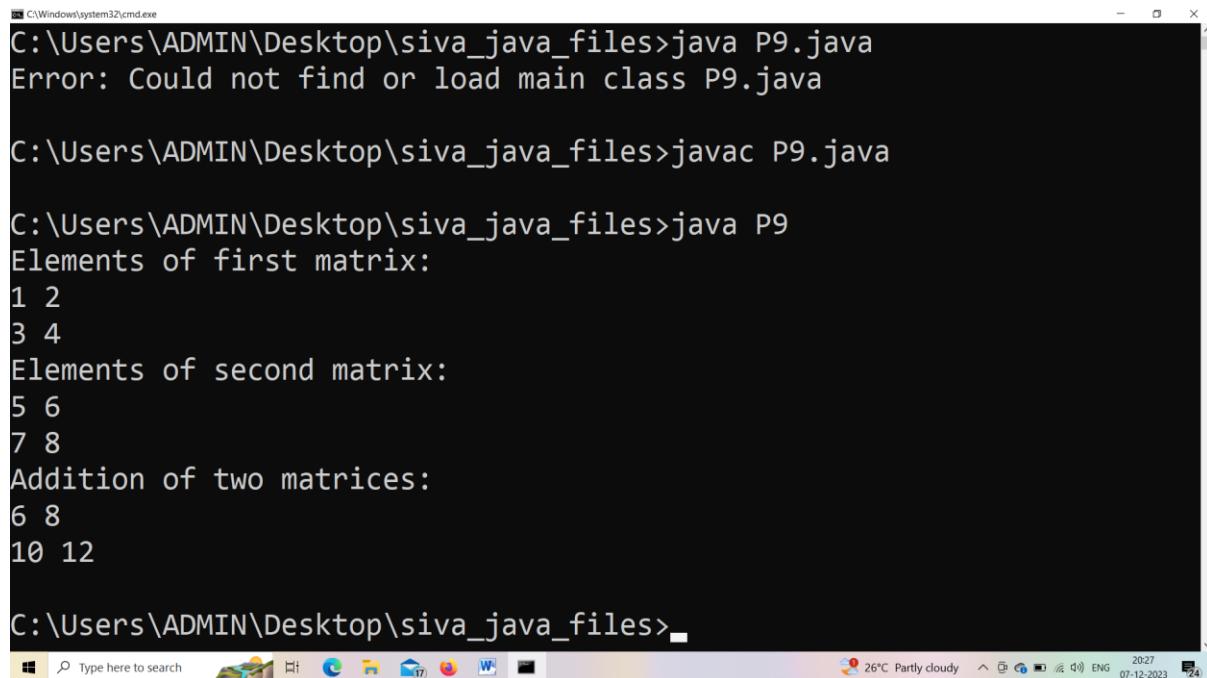
9. Write a program to print elements in matrix form.

Program:

```
class P9
{
    public static void main(String args[])
    {
        int a[][]=new int[2][2];
        int b[][]=new int[2][2];
        a[0][0]=1;
        a[0][1]=2;
        a[1][0]=3;
        a[1][1]=4;
        b[0][0]=5;
        b[0][1]=6;
        b[1][0]=7;
        b[1][1]=8;
        int c[][]=new int[2][2];
        System.out.println("Elements of first matrix:");
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                System.out.print(a[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println("Elements of second matrix:");
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                System.out.print(b[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println("Addition of two matrices:");
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
```

```
{  
System.out.print(c[i][j]+" ");  
}  
System.out.println();  
}  
}  
}
```

Output:



C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>java P9.java
Error: Could not find or load main class P9.java

C:\Users\ADMIN\Desktop\siva_java_files>javac P9.java

C:\Users\ADMIN\Desktop\siva_java_files>java P9
Elements of first matrix:
1 2
3 4
Elements of second matrix:
5 6
7 8
Addition of two matrices:
6 8
10 12

C:\Users\ADMIN\Desktop\siva_java_files>

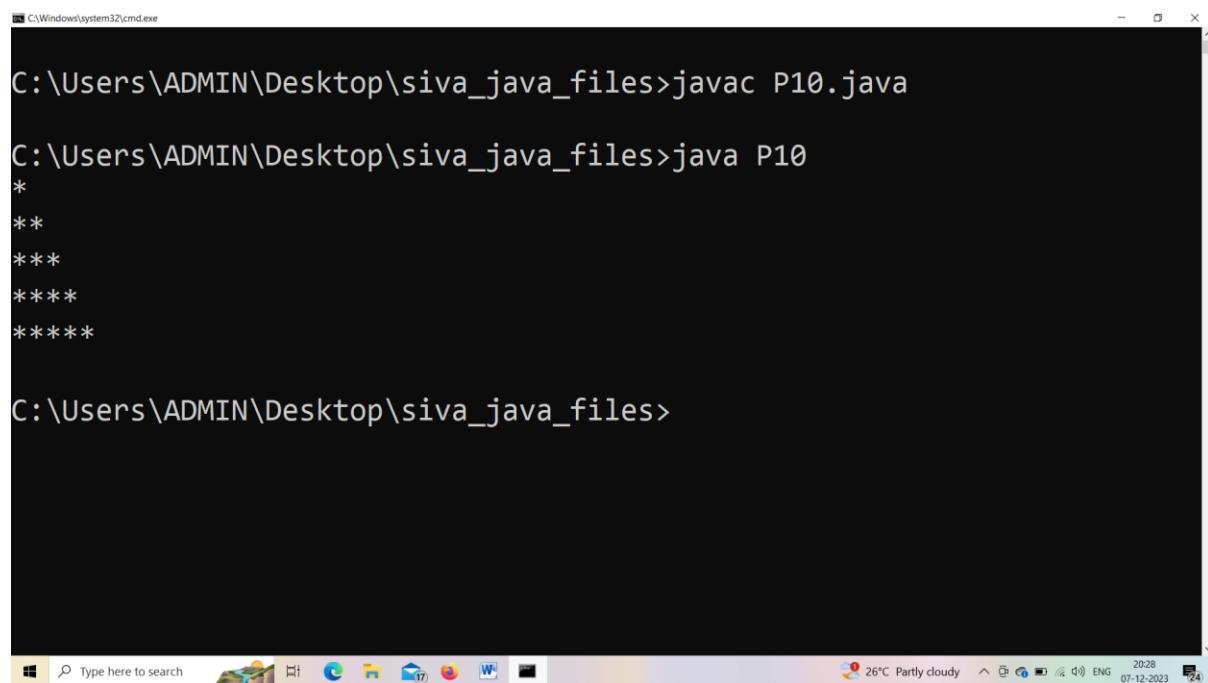
The screenshot shows a Windows Command Prompt window titled 'cmd.exe'. The command 'javac P9.java' is run, but it fails to find the main class. Then, 'java P9' is run, which executes the code. It prints 'Elements of first matrix:' followed by the elements [1, 2] and [3, 4]. It then prints 'Elements of second matrix:' followed by the elements [5, 6] and [7, 8]. Finally, it prints 'Addition of two matrices:' followed by the resulting matrix [[6, 8], [10, 12]]. The taskbar at the bottom shows various icons and the system tray with weather information (26°C Partly cloudy) and date/time (07-12-2023).

10. Write a program to print starts in increasing order.

Program:-

```
class P10
{
    public static void main(String args[])
    {
        int n=5;
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=i;j++)
            {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The user has typed 'javac P10.java' to compile the Java file. After compilation, they run 'java P10', which outputs five lines of asterisks ('*') of increasing length, representing a right-angled triangle. The command prompt then returns to the directory 'C:\Users\ADMIN\Desktop\siva_java_files>'.

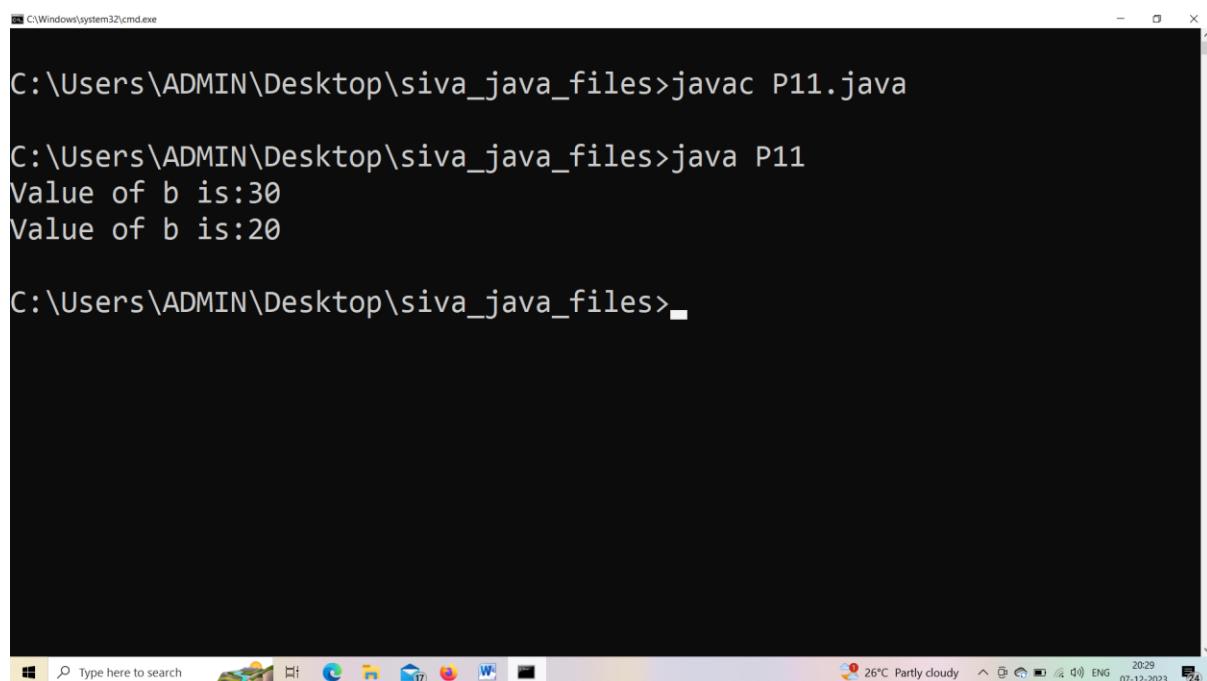
```
C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac P10.java
C:\Users\ADMIN\Desktop\siva_java_files>java P10
*
**
***
****
*****
```

11. Write a Demo Program On Conditional Operator.

Program:-

```
public class P11
{
    public static void main(String args[])
    {
        int a,b;
        a=10;
        b=(a==1)?20:30;
        System.out.println("Value of b is:"+b);
        b=(a==10)?20:30;
        System.out.println("Value of b is:"+b);
    }
}
```

Output:-



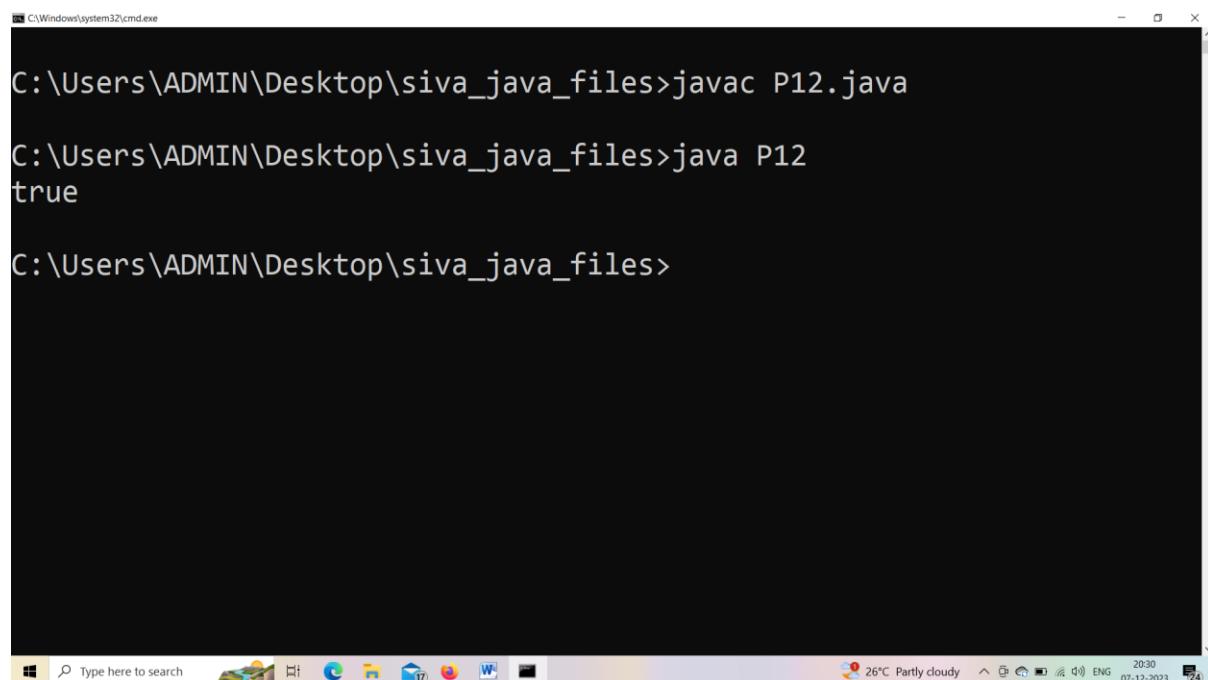
```
C:\Users\ADMIN\Desktop\siva_java_files>javac P11.java
C:\Users\ADMIN\Desktop\siva_java_files>java P11
Value of b is:30
Value of b is:20
C:\Users\ADMIN\Desktop\siva_java_files>
```

12. Write a program for instance of operator.

Program:-

```
public class P12
{
    public static void main(String args[])
    {
        String name="James";
        boolean result=name instanceof String;
        System.out.println(result);
    }
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac P12.java
C:\Users\ADMIN\Desktop\siva_java_files>java P12
true

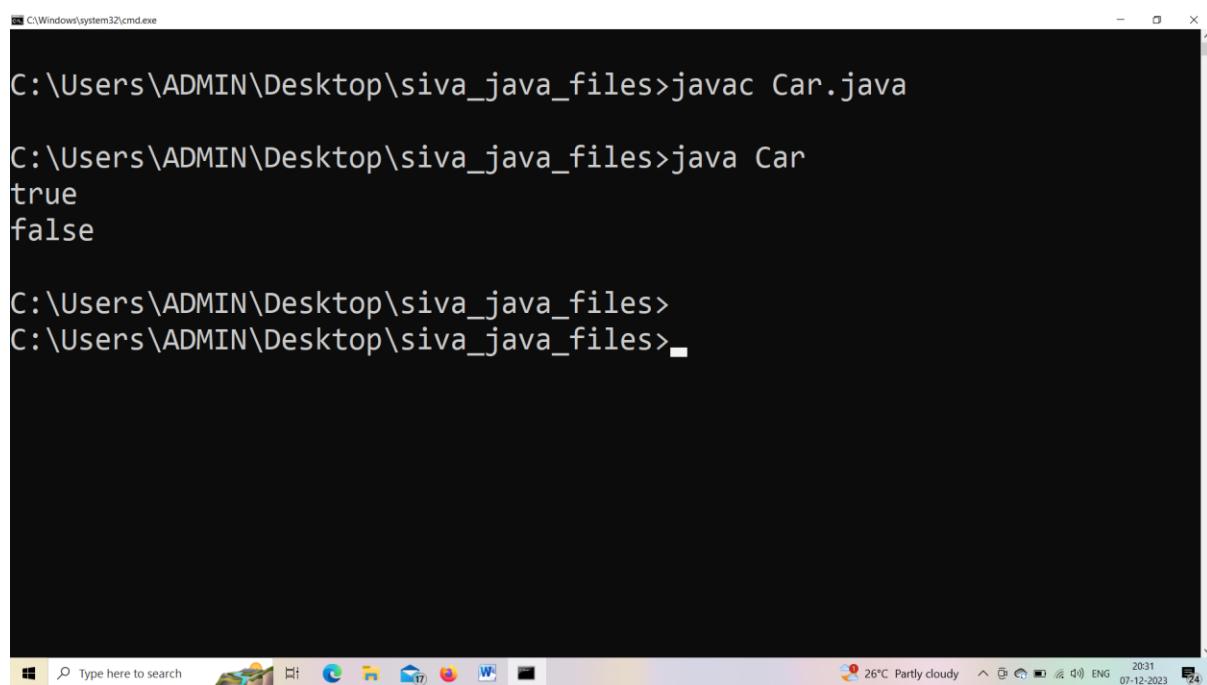
C:\Users\ADMIN\Desktop\siva_java_files>
```

13. Write a demo program on Inheritance.

Program:-

```
class Vehicle
{
}
public class Car extends Vehicle
{
    public static void main(String args[])
    {
        Vehicle ve=new Car();
        boolean result=ve instanceof Car;
        System.out.println(result);
        Object obj = new String("hello");
        boolean set = obj instanceof Integer;
        System.out.println(set);
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac Car.java' is run, followed by 'java Car'. The output shows 'true' on the first line and 'false' on the second line, demonstrating that the Car object is an instance of the Vehicle class but not of the Integer class. The taskbar at the bottom shows various application icons like File Explorer, Edge, and File Manager.

```
C:\Users\ADMIN\Desktop\siva_java_files>javac Car.java
C:\Users\ADMIN\Desktop\siva_java_files>java Car
true
false
```

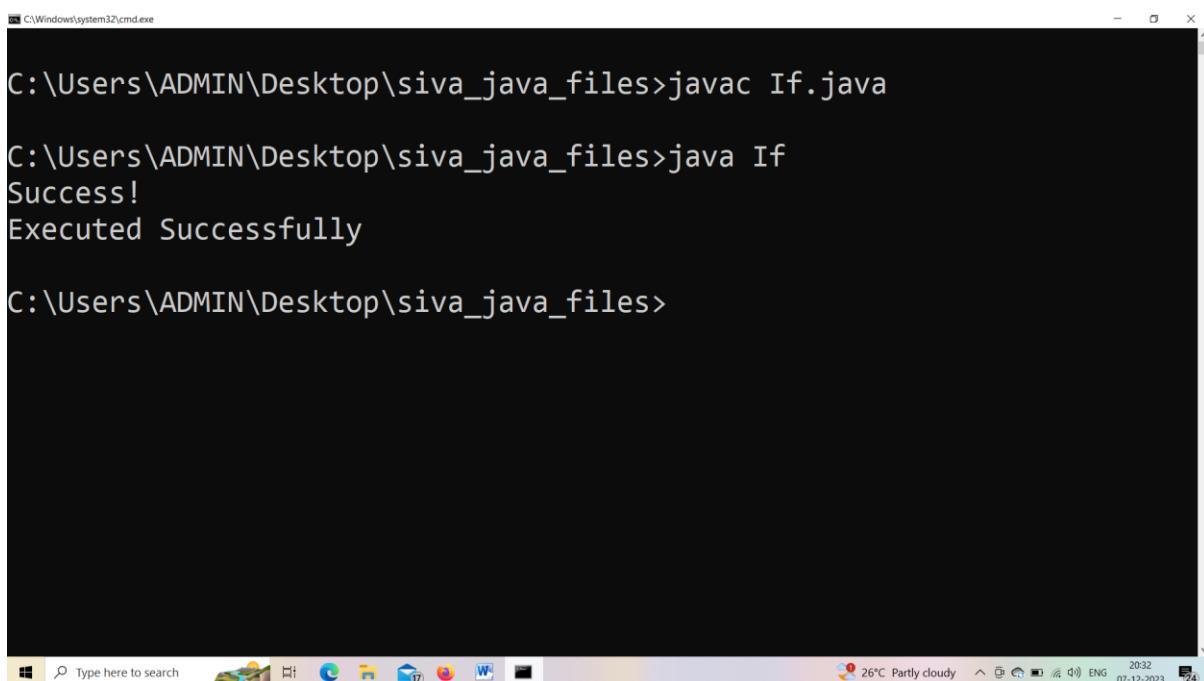
14. Write a demo program on If statement.

Program:-

```
class If
{
public static void main(String args[])
{
int test=10;

if(test>6){
System.out.println("Success!");
}
System.out.println("Executed Successfully");
}
}
```

Output:-



```
C:\Windows\System32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac If.java
C:\Users\ADMIN\Desktop\siva_java_files>java If
Success!
Executed Successfully

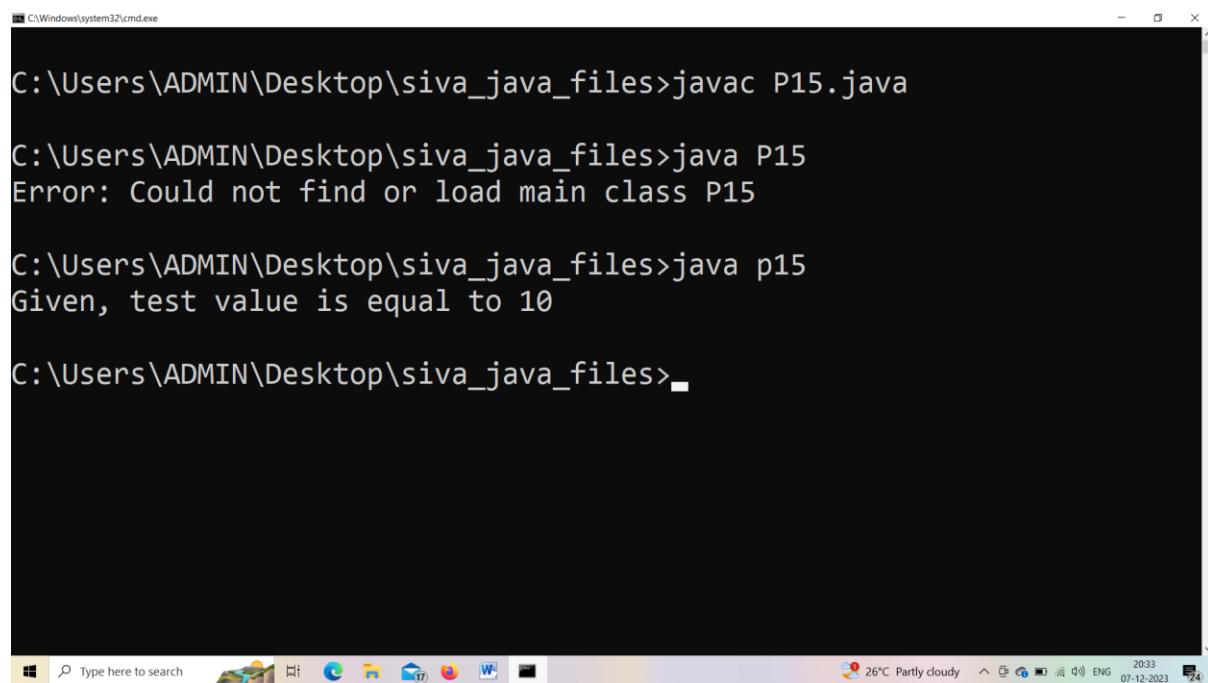
C:\Users\ADMIN\Desktop\siva_java_files>
```

15. Write a Demo Program on IF-ELSE Statement

Program:-

```
class p15
{
    public static void main(String args[])
    {
        int test = 10;
        if(test == 10)
        {
            System.out.println("Given, test value is equal to 10");
        }
        else
        {
            System.out.println("test value is not equal to 10");
        }
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command line shows the following sequence of operations:

- javac P15.java
- java P15
- Given, test value is equal to 10
- java p15

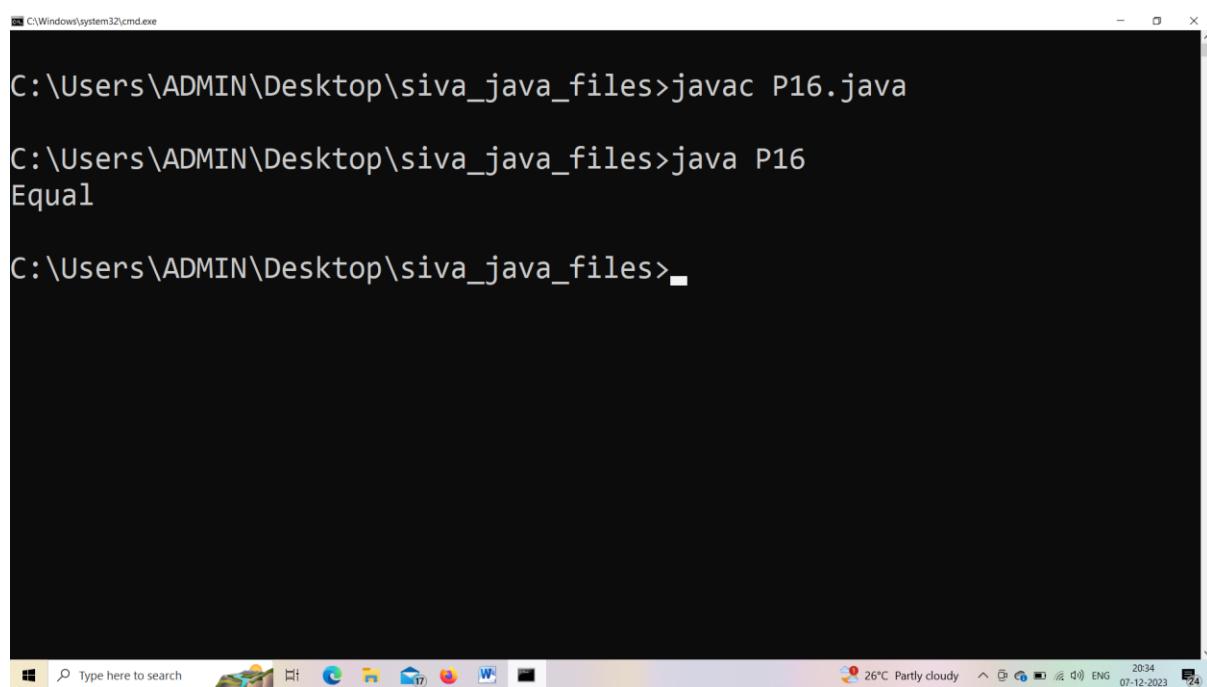
The window includes standard Windows taskbar icons at the bottom and system status information in the bottom right corner.

16. Write a Program On Nested IF-ELSE.

Program:-

```
public class P16
{
    public static void main(String args[]){
        int a = 2,b = 2,c = 2;
        if(a==b)
        {
            if(a==c)
            {
                System.out.println("Equal");
            }
            else
            {
                System.out.println("Not Equal");
            }
        }
        else
        {
            System.out.println("Not Equal");
        }
    }
}
```

Output:-



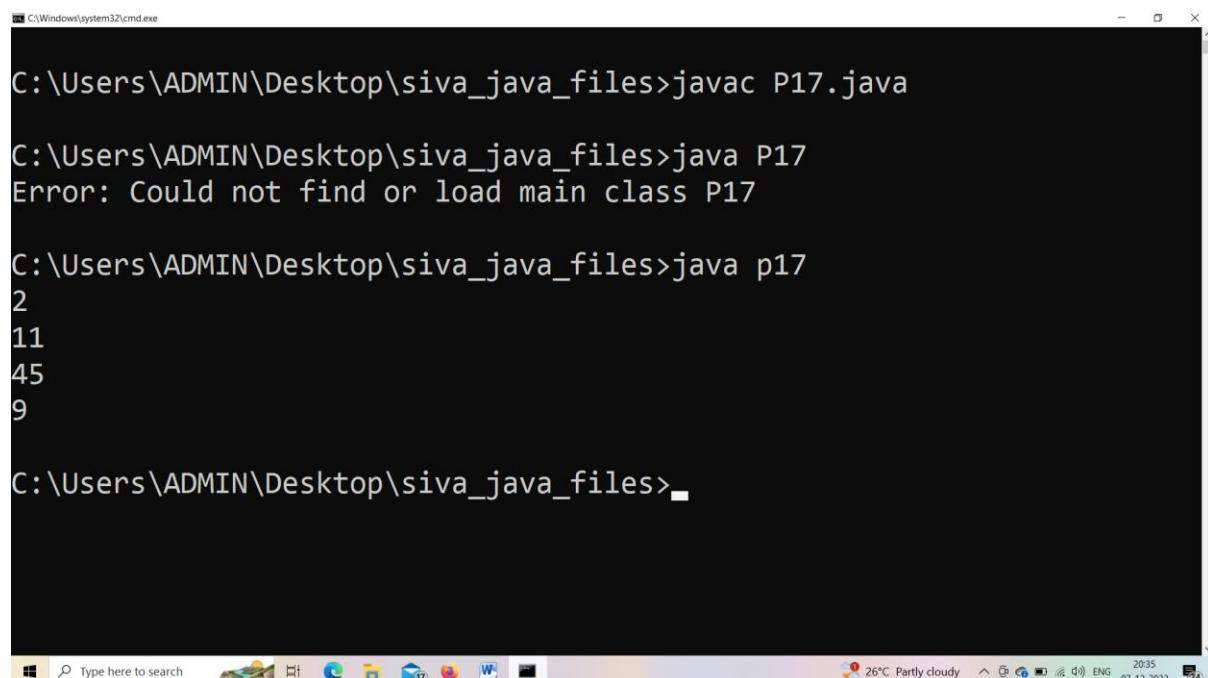
```
C:\Users\ADMIN\Desktop\siva_java_files>javac P16.java
C:\Users\ADMIN\Desktop\siva_java_files>java P16
Equal
C:\Users\ADMIN\Desktop\siva_java_files>
```

17. Write a program on While loop.

Program:-

```
class p17 {
public static void main(String args[])
{
int arr[]={2,11,45,9};
int i=0;
while(i<4)
{
System.out.println(arr[i]+" ");
i++;
}
}
}
```

Output:-



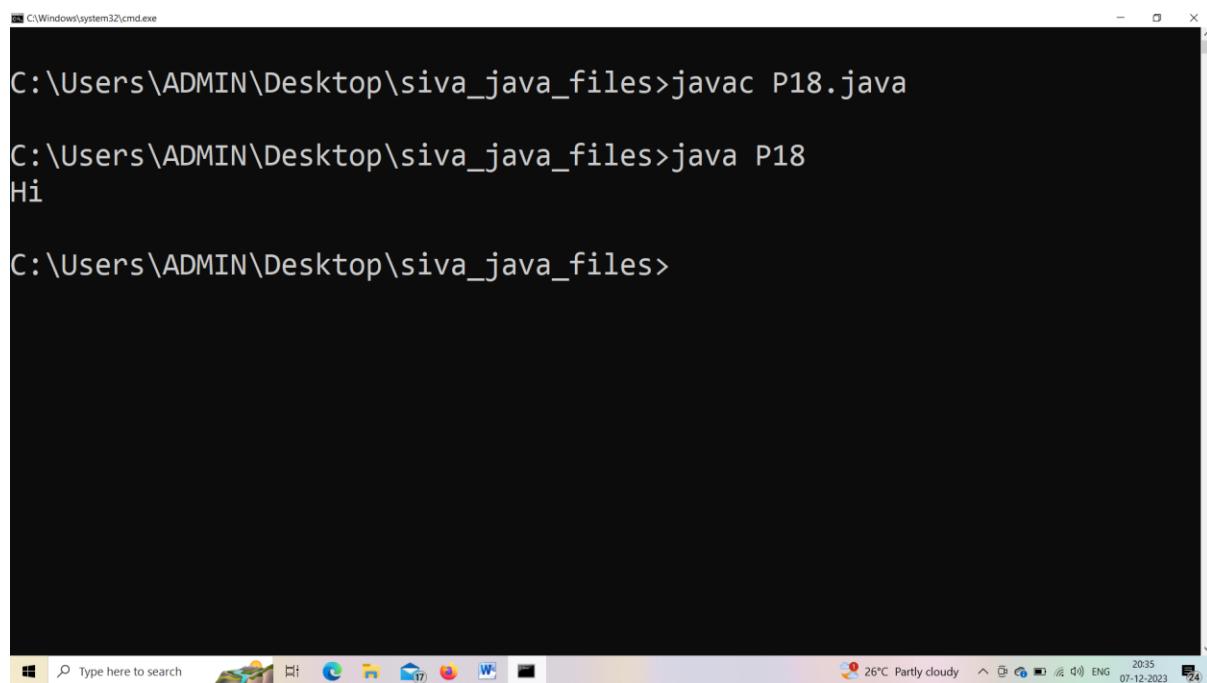
```
C:\Users\ADMIN\Desktop\siva_java_files>javac P17.java
C:\Users\ADMIN\Desktop\siva_java_files>java P17
Error: Could not find or load main class P17
C:\Users\ADMIN\Desktop\siva_java_files>java p17
2
11
45
9
C:\Users\ADMIN\Desktop\siva_java_files>
```

18. Write a demo program on If-else ladder.

Program:-

```
public class P18
{
    public static void main(String args[])
    {
        int test=2;
        if(test==1)
            System.out.println("Hello");
        else if(test==2)
            System.out.println("Hi");
        else if(test==3)
            System.out.println("Good");
        else
            System.out.println("No match found");
    }
}
```

Output:-



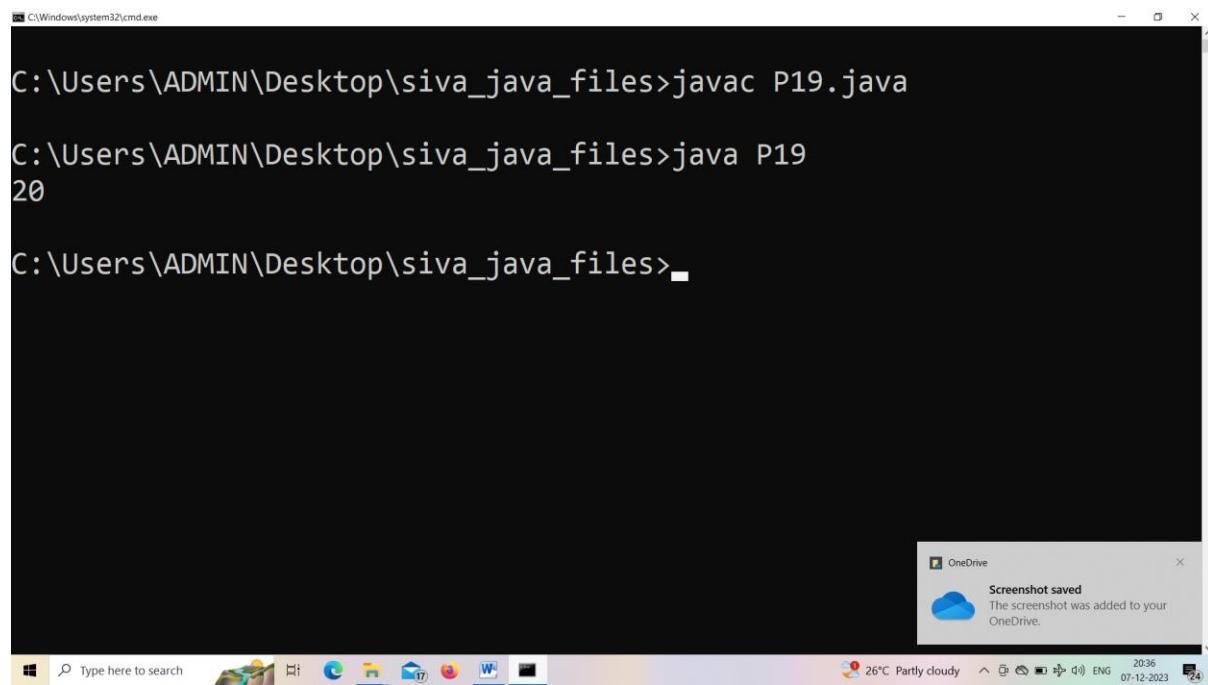
```
C:\Users\ADMIN\Desktop\siva_java_files>javac P18.java
C:\Users\ADMIN\Desktop\siva_java_files>java P18
Hi
C:\Users\ADMIN\Desktop\siva_java_files>
```

19. Write a demo program on DoWhile.

Program:-

```
class P19
{
    public static void main(String args[])
    {
        int i=20;
        do
        {
            System.out.println(i);
            i--;
        }
        while(i<1);
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'cmd.exe' with the following output:

```
C:\Users\ADMIN\Desktop\siva_java_files>javac P19.java
C:\Users\ADMIN\Desktop\siva_java_files>java P19
20
```

The Java program outputs the value 20, which is the initial value of the variable 'i' before the loop begins. The command prompt then returns to the directory 'siva_java_files'.

In the bottom right corner of the screen, there is a notification from OneDrive stating 'Screenshot saved' with the message 'The screenshot was added to your OneDrive.' This indicates that the screenshot of the command prompt was taken and stored in the user's OneDrive account.

20. Write a program on Switch Case.

Program:-

```
class P20
{
    public static void main(String args[])
    {
        int month=2;
        int year=2000;
        int numOfDays=0;
        switch(month)
        {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                numOfDays=31;
                break;
            case 2:
                if((year%4==0 && year%100!=0)||(year%400==0))
                    numOfDays=29;
                else
                    numOfDays=28;
                break;
            default:
                System.out.println("Invalid month");
                break;
        }
        System.out.println("Number of days is:"+numOfDays);
    }
}
```

Output:-

```
C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac P20.java
C:\Users\ADMIN\Desktop\siva_java_files>java P20
Number of days is:29

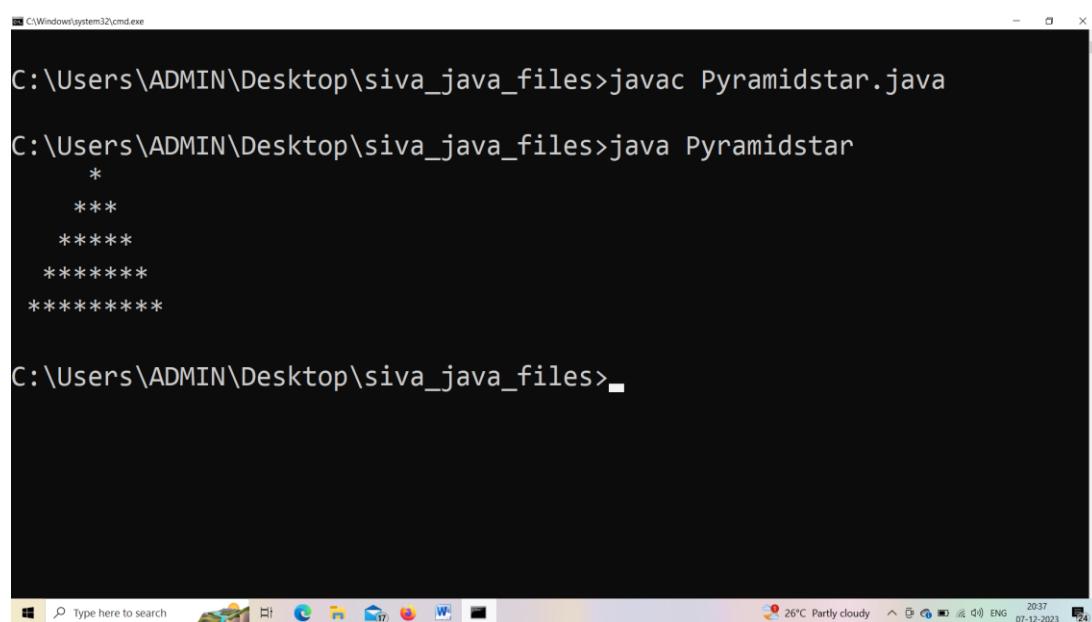
C:\Users\ADMIN\Desktop\siva_java_files>
```

21. Write a program to implement Pyramid Star Pattern.

Program:-

```
class Pyramidstar
{
    public static void main(String args[])
    {
        int n=5;
        for(int i=1;i<=n;i++)
        {
            for(int j=i;j<=n;j++)
            {
                System.out.print(" ");
            }
            for(int j=1;j<i;j++)
            {
                System.out.print("*");
            }
            for(int j=1;j<=i;j++)
            {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac Pyramidstar.java' is entered and executed, followed by 'java Pyramidstar'. The output displays a pyramid star pattern with 5 rows of asterisks (*). The taskbar at the bottom shows various application icons, and the system tray indicates the date as 07-12-2023 and the time as 20:37.

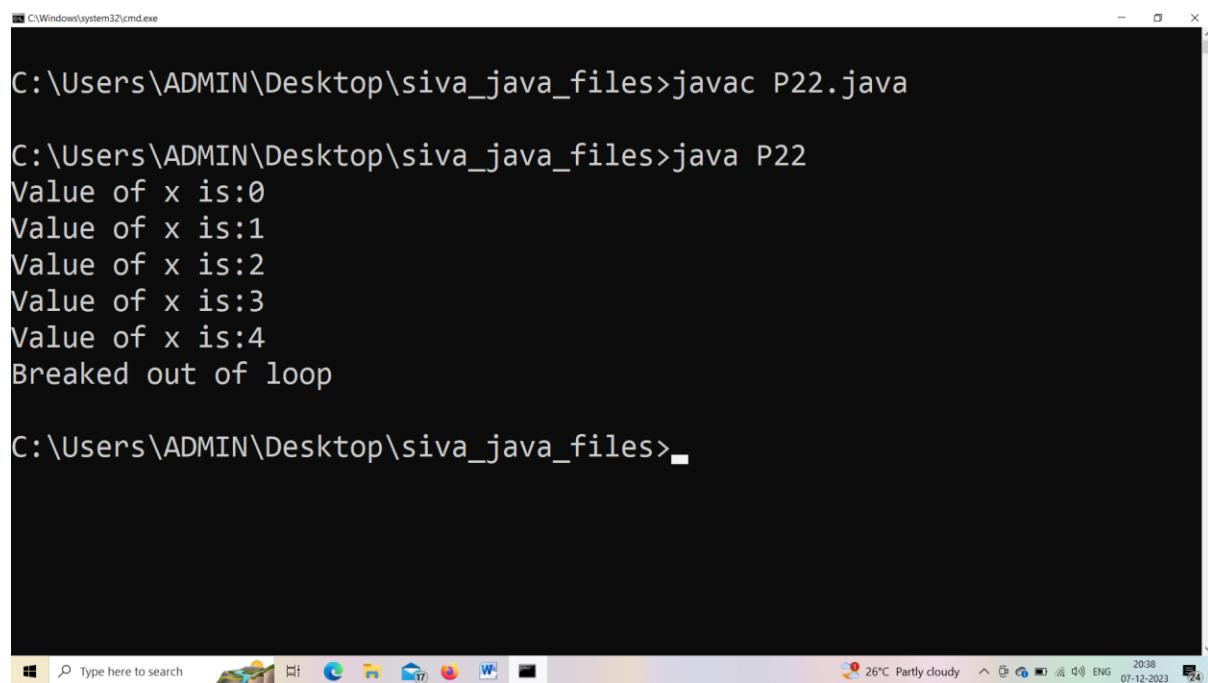
```
C:\Users\ADMIN\Desktop\siva_java_files>javac Pyramidstar.java
C:\Users\ADMIN\Desktop\siva_java_files>java Pyramidstar
*
 ***
 ****
 *****
 ******
 *****
C:\Users\ADMIN\Desktop\siva_java_files>
```

22. Write a program on Break Statement.

Program:-

```
class P22
{
    public static void main(String args[])
    {
        for(int x=0;x<50;x++)
        {
            if(x==5)
            {
                break;
            }
            System.out.println("Value of x is:"+x);
        }
        System.out.println("Breaked out of loop");
    }
}
```

Output:-



```
C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac P22.java

C:\Users\ADMIN\Desktop\siva_java_files>java P22
Value of x is:0
Value of x is:1
Value of x is:2
Value of x is:3
Value of x is:4
Breaked out of loop

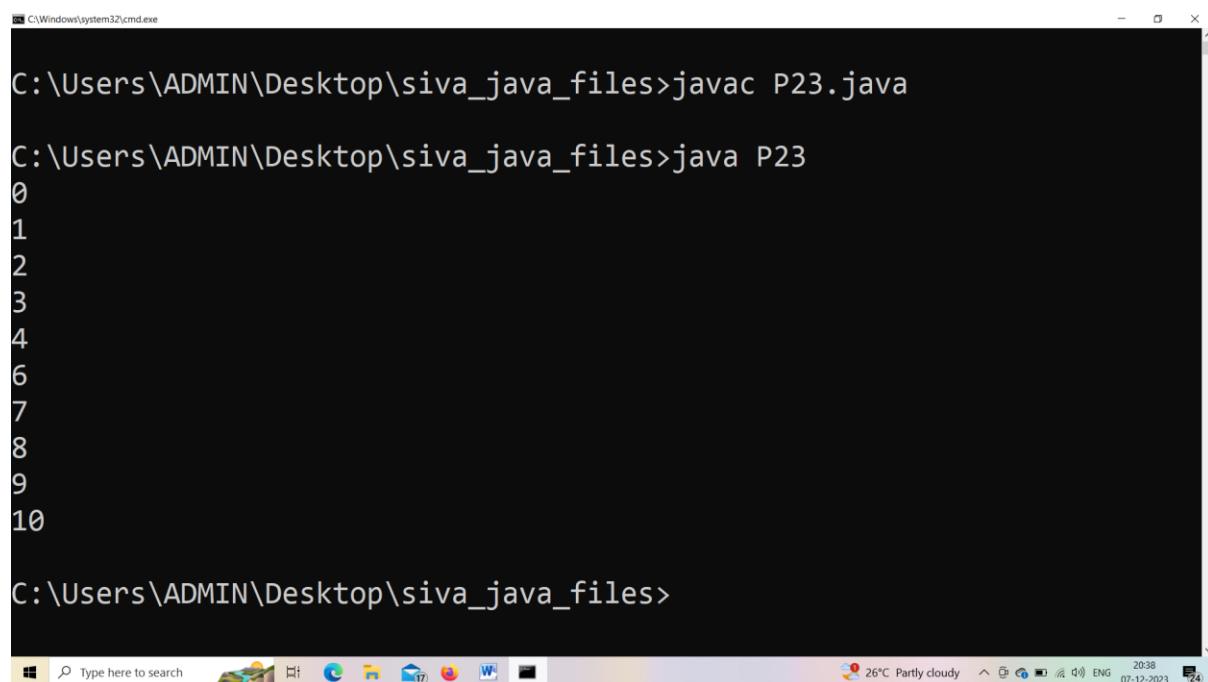
C:\Users\ADMIN\Desktop\siva_java_files>
```

23. Write a program on Continue Statement.

Program:-

```
class P23
{
    public static void main(String args[])
    {
        int i;
        for(i=0;i<=10;i++)
        {
            if(i==5)
                continue;
            System.out.println(i);
        }
    }
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac P23.java
C:\Users\ADMIN\Desktop\siva_java_files>java P23
0
1
2
3
4
6
7
8
9
10

C:\Users\ADMIN\Desktop\siva_java_files>
```

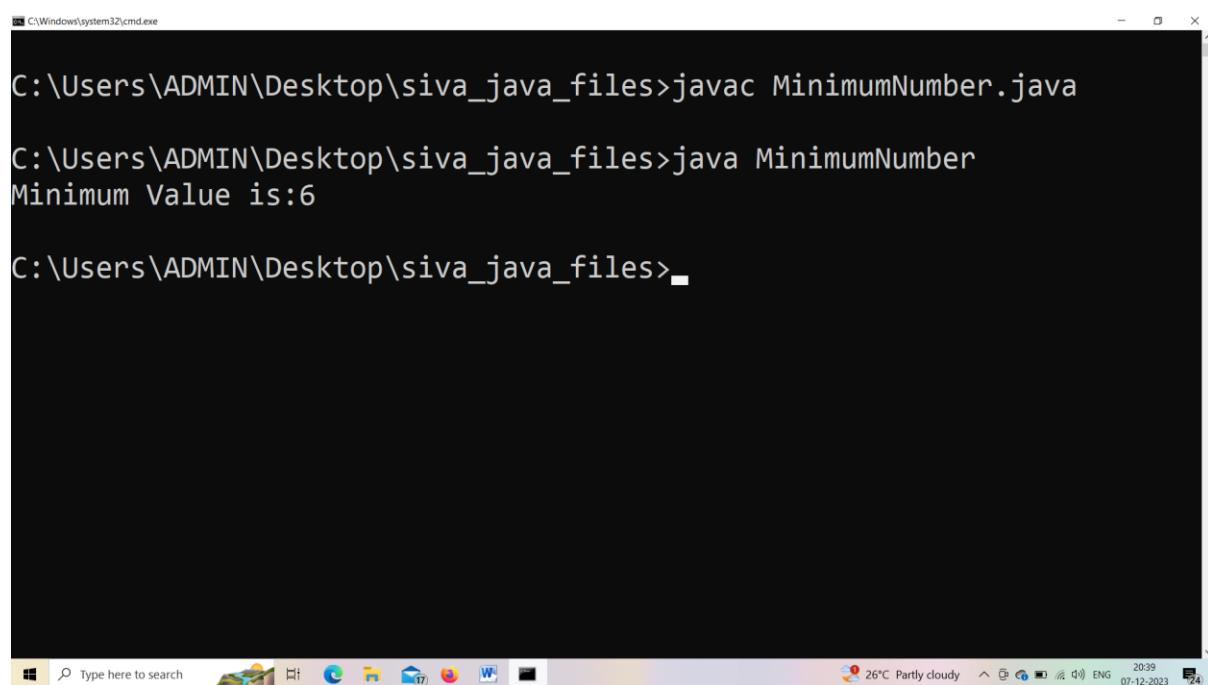
24. Write a program to find Minimum Number Using Functions.

Program:-

```
public class MinimumNumber
{
    public static void main(String args[])
    {
        int a=11;
        int b=6;
        int c=minFunction(a,b);
        System.out.println("Minimum Value is:" +c);
    }

    public static int minFunction(int n1, int n2){
        int min;
        if(n1 > n2)
            min = n2;
        else
            min = n1;
        return min;
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The command 'javac MinimumNumber.java' is entered and executed, followed by 'java MinimumNumber', which outputs 'Minimum Value is:6'. The taskbar at the bottom shows various icons for applications like File Explorer, Edge, Mail, and File History.

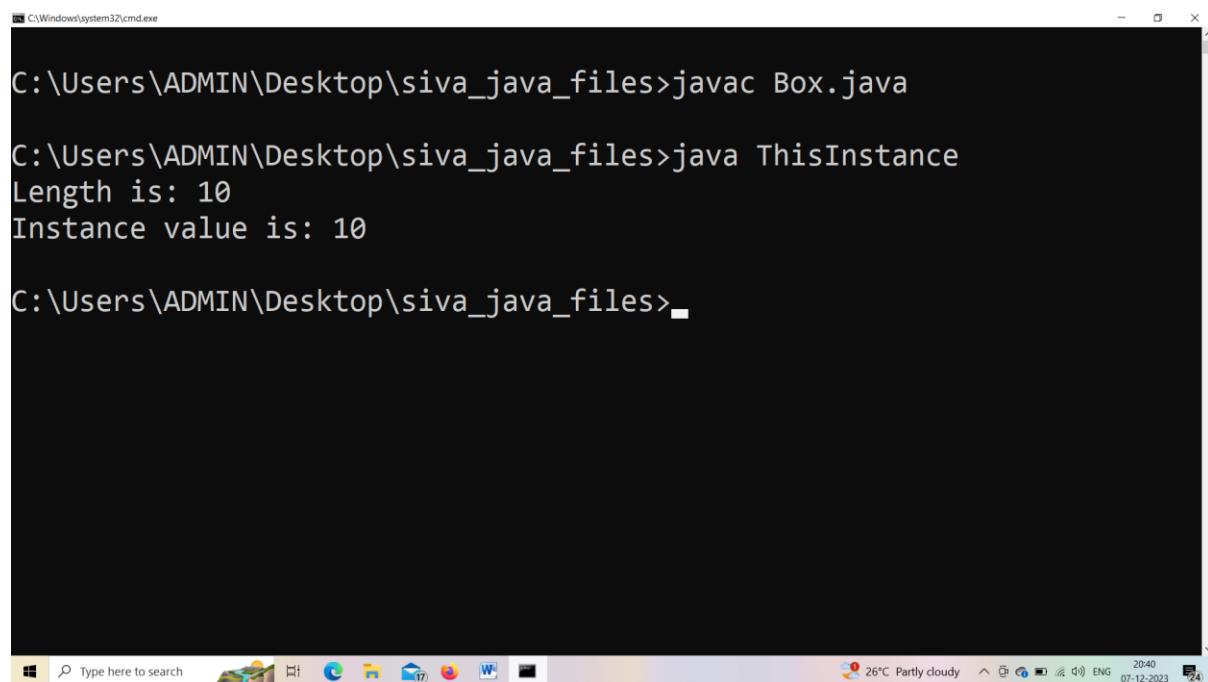
```
C:\Users\ADMIN\Desktop\siva_java_files>javac MinimumNumber.java
C:\Users\ADMIN\Desktop\siva_java_files>java MinimumNumber
Minimum Value is:6
C:\Users\ADMIN\Desktop\siva_java_files>
```

25. Write a program on THIS for instance variable.

Program:-

```
class Box
{
int len = 10;
void meth() {
int len = 40;
System.out.println("Length is:"+len);
System.out.println("Instance value is:"+this.len);
}
}
class ThisInstance
{
public static void main(String args[])
{
Box obj = new Box ();
obj.meth();
}
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The command 'javac Box.java' is run, followed by 'java ThisInstance'. The output shows the length as 10 and the instance value as 10. The taskbar at the bottom includes icons for File Explorer, Edge, Mail, and File Explorer again, along with system status icons like battery level and network.

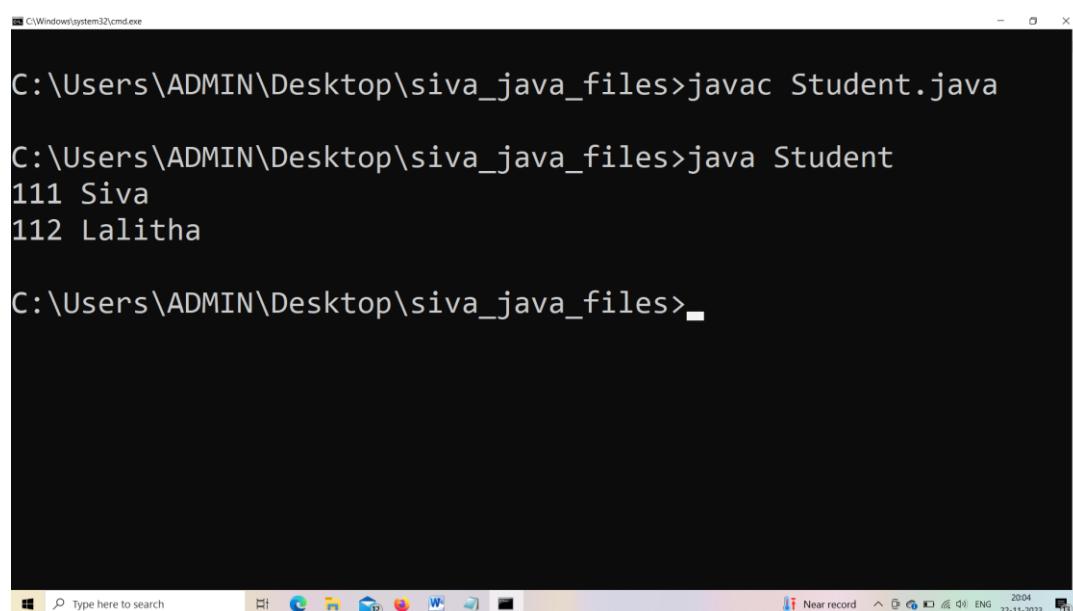
```
C:\Users\ADMIN\Desktop\siva_java_files>javac Box.java
C:\Users\ADMIN\Desktop\siva_java_files>java ThisInstance
Length is: 10
Instance value is: 10
```

26. Write a program on usage of class and new keyword.

Program:-

```
class Student
{
    int rollno;
    String name;
    void insertRecord(int r, String n)
    {
        rollno = r;
        name = n;
    }
    void displayInformation()
    {
        System.out.println(rollno+" "+name);
    }
    public static void main(String args[])
    {
        Student s1 = new Student();
        Student s2 = new Student();
        s1.insertRecord(111,"Siva");
        s2.insertRecord(112,"Lalitha");
        s1.displayInformation();
        s2.displayInformation();
    }
}
```

Output:-



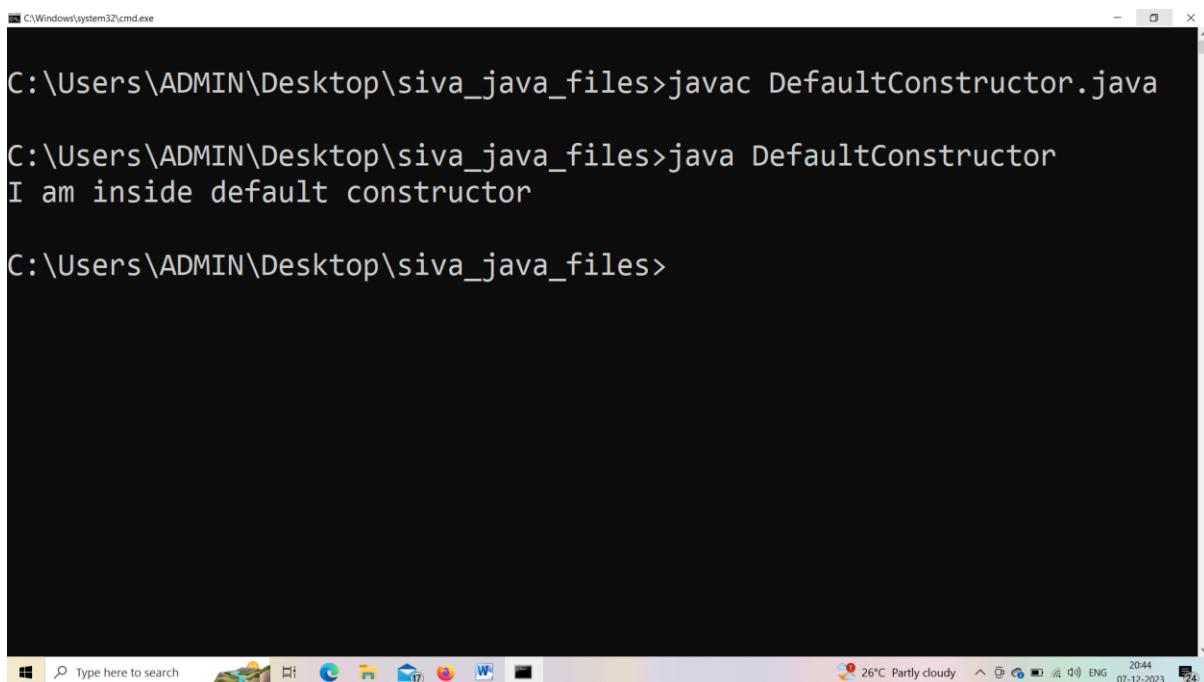
```
C:\Users\ADMIN\Desktop\siva_java_files>javac Student.java
C:\Users\ADMIN\Desktop\siva_java_files>java Student
111 Siva
112 Lalitha
```

27. Write a program on a Default Constructor.

Program:-

```
public class DefaultConstructor
{
    public DefaultConstructor()
    {
        System.out.println("I am inside default constructor");
    }
    public static void main(String args[])
    {
        DefaultConstructor dc = new DefaultConstructor();
    }
}
```

Output:-



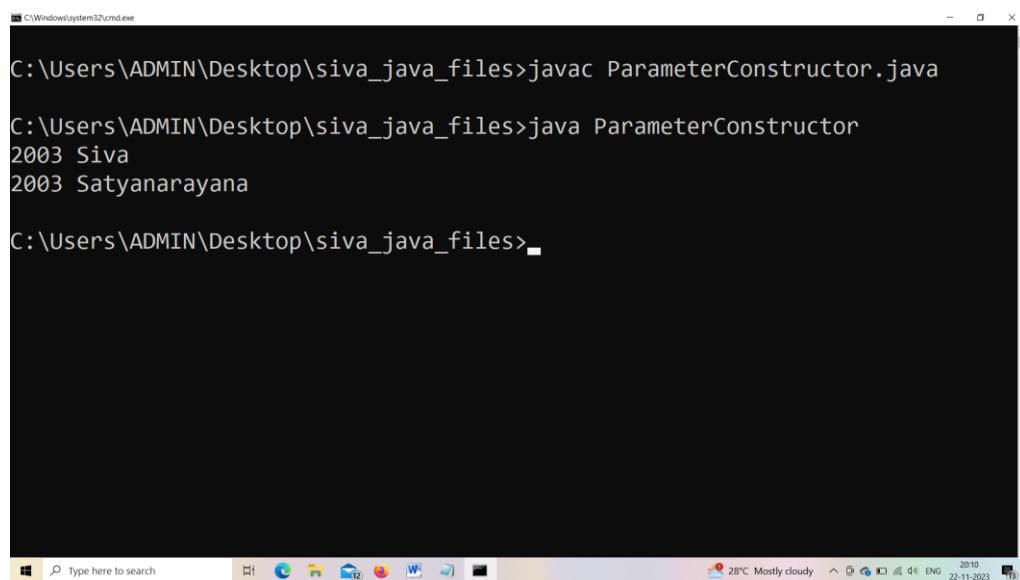
```
C:\Users\ADMIN\Desktop\siva_java_files>javac DefaultConstructor.java
C:\Users\ADMIN\Desktop\siva_java_files>java DefaultConstructor
I am inside default constructor
C:\Users\ADMIN\Desktop\siva_java_files>
```

28. Write a Program on Parameter Constructor.

Program:-

```
class ParameterConstructor
{
int id;
String name;
ParameterConstructor(int i, String n)
{
id = i;
name = n;
}
void display()
{
System.out.println(id+" "+name);
}
public static void main(String args[])
{
ParameterConstructor pc1 = new ParameterConstructor(2003,"Siva");
ParameterConstructor pc2 = new ParameterConstructor(2003,"Satyanarayana");
pc1.display();
pc2.display();
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac ParameterConstructor.java
C:\Users\ADMIN\Desktop\siva_java_files>java ParameterConstructor
2003 Siva
2003 Satyanarayana
```

29. Write a program on Constructor Overloading.

Program:-

```
class ConstructorOverloading
{
    int id;
    String name;
    int age;
    ConstructorOverloading (int i, String n)
    {
        id = i;
        name = n;
    }
    ConstructorOverloading (int i, String n, int a)
    {
        id = i;
        name = n;
        age = a;
    }
    void display()
    {
        System.out.println(id+" "+name+" "+age);
    }
    public static void main(String args[])
    {
        ConstructorOverloading co1 = new ConstructorOverloading(2003,"Siva",20);
        ConstructorOverloading co2 = new ConstructorOverloading(2003,"Anu");
        co1.display();
        co2.display();
    }
}
```

Output:-

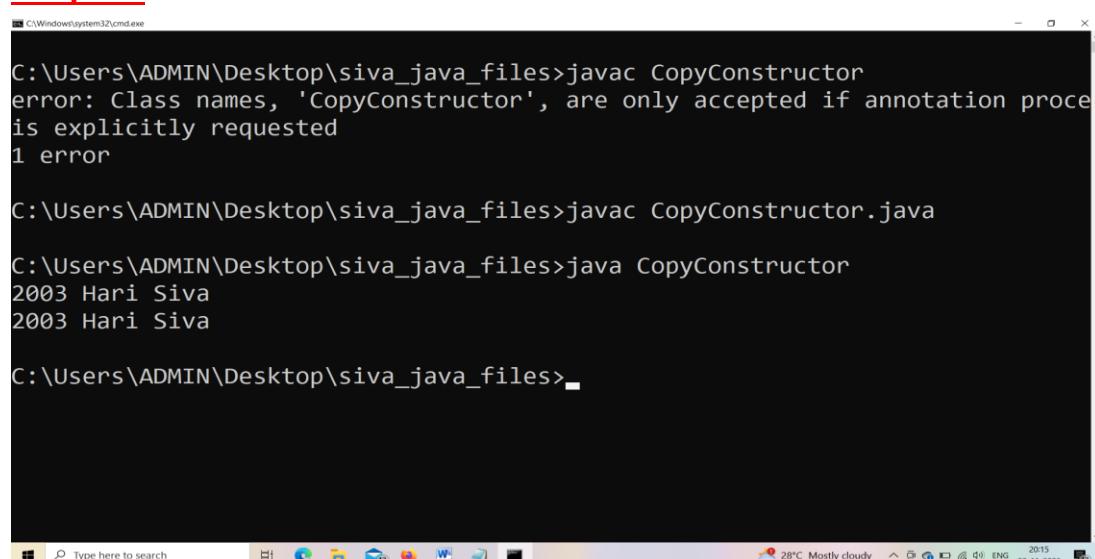
```
C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac ConstructorOverloading.java
C:\Users\ADMIN\Desktop\siva_java_files>java ConstructorOverloading
2003 Siva 20
2003 Anu 0
C:\Users\ADMIN\Desktop\siva_java_files>
```

30. Write a program on Copy Constructor.

Program:-

```
class CopyConstructor
{
int id;
String name;
CopyConstructor(int i, String n)
{
id = i;
name = n;
}
CopyConstructor(CopyConstructor s)
{
id = s.id;
name = s.name;
}
void display()
{
System.out.println(id+" "+name);
}
public static void main(String args[])
{
CopyConstructor cc1 = new CopyConstructor(2003,"Hari Siva");
CopyConstructor cc2 = new CopyConstructor(cc1);
cc1.display();
cc2.display();
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac CopyConstructor' is run, resulting in an error message: 'error: Class names, 'CopyConstructor', are only accepted if annotation process is explicitly requested'. One error is reported. Subsequent commands 'javac CopyConstructor.java' and 'java CopyConstructor' are run successfully, outputting '2003 Hari Siva' twice. The taskbar at the bottom shows various pinned icons and the system tray indicates it's 20:15, 22-11-2023, with a weather icon showing 28°C Mostly cloudy.

```
C:\Users\ADMIN\Desktop\siva_java_files>javac CopyConstructor
error: Class names, 'CopyConstructor', are only accepted if annotation process
is explicitly requested
1 error

C:\Users\ADMIN\Desktop\siva_java_files>javac CopyConstructor.java

C:\Users\ADMIN\Desktop\siva_java_files>java CopyConstructor
2003 Hari Siva
2003 Hari Siva

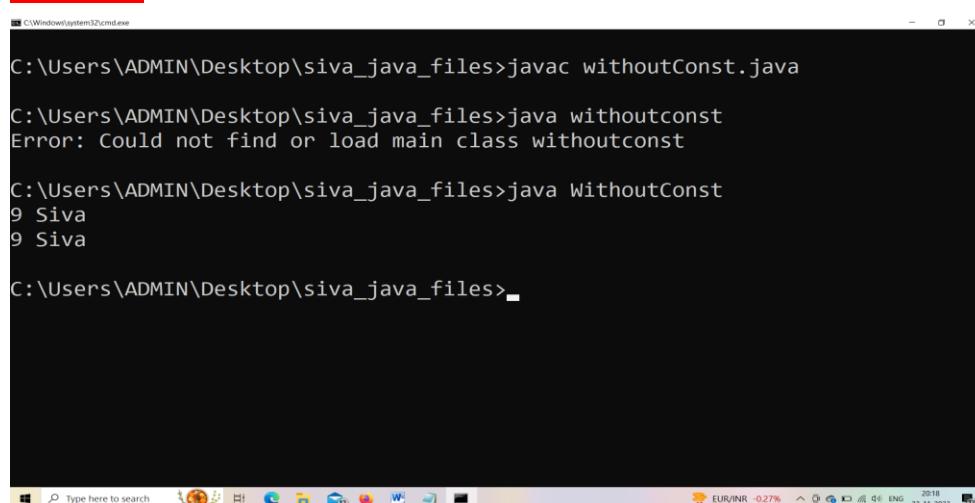
C:\Users\ADMIN\Desktop\siva_java_files>
```

31. Write a program on Copying an Object Without Constructor.

Program:-

```
class WithoutConst
{
int id;
String name;
WithoutConst(int i, String n)
{
    id = i;
    name = n;
}
WithoutConst()
{
}
void display()
{
    System.out.println(id+" "+name);
}
public static void main(String args[])
{
    WithoutConst wc1 = new WithoutConst(9, "Siva");
    WithoutConst wc2 = new WithoutConst();
    wc2.id = wc1.id;
    wc2.name=wc1.name;
    wc1.display();
    wc2.display();
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac withoutConst.java' is run, followed by 'java withoutconst'. An error message 'Error: Could not find or load main class withoutconst' is displayed. Finally, 'java WithoutConst' is run, outputting '9 Siva' twice. The taskbar at the bottom shows various icons and system status.

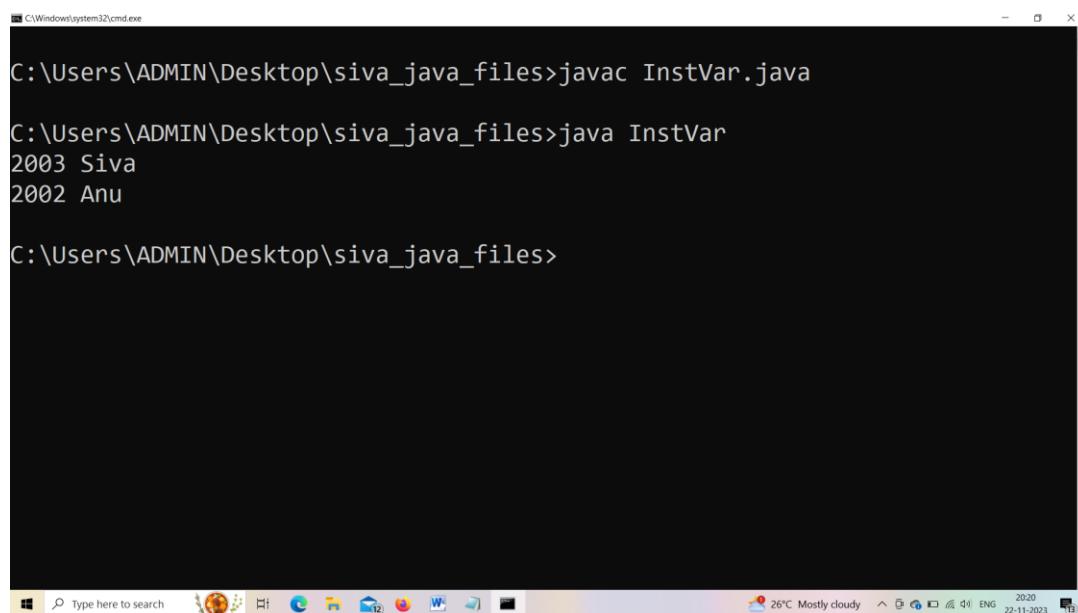
```
C:\Users\ADMIN\Desktop\siva_java_files>javac withoutConst.java
C:\Users\ADMIN\Desktop\siva_java_files>java withoutconst
Error: Could not find or load main class withoutconst
C:\Users\ADMIN\Desktop\siva_java_files>java WithoutConst
9 Siva
9 Siva
C:\Users\ADMIN\Desktop\siva_java_files>
```

32. Write a Program to access the Instance Variable using THIS keyword

Program:-

```
class InstVar
{
    int id;
    String name;
    InstVar(int id, String name)
    {
        this.id=id;
        this.name=name;
    }
    void display()
    {
        System.out.println(id+ " "+name);
    }
    public static void main (String args[])
    {
        InstVar iv1 = new InstVar(2003,"Siva");
        InstVar iv2 = new InstVar(2002,"Anu");
        iv1.display();
        iv2.display();
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac InstVar.java' is entered and executed, followed by the command 'java InstVar'. The output displays two lines of text: '2003 Siva' and '2002 Anu', indicating the values of the instance variables for each object.

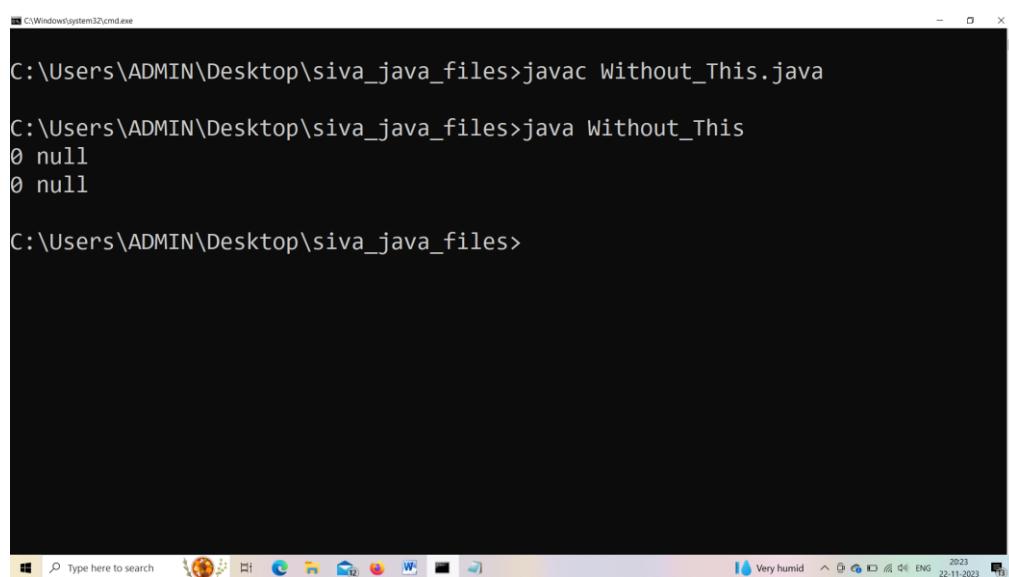
```
C:\Users\ADMIN\Desktop\siva_java_files>javac InstVar.java
C:\Users\ADMIN\Desktop\siva_java_files>java InstVar
2003 Siva
2002 Anu
```

33. Write a program to access Instance Variables of a class Without THIS keyword.

Program:-

```
class Without_This
{
int id;
String name;
Without_This (int id, String name)
{
id = id;
name = name;
}
void display ()
{
System.out.println(id+" "+name);
}
public static void main(String args[])
{
Without_This wt1 = new Without_This (2003,"Siva");
Without_This wt2 = new Without_This (2003, "Veera");
wt1.display();
wt2.display();
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac Without_This.java' is entered and executed, followed by 'java Without_This'. The output shows two lines of text: '0 null' and '0 null', indicating the values of the instance variables id and name respectively. The taskbar at the bottom of the screen includes icons for File Explorer, Task View, Mail, Edge, and File Explorer again.

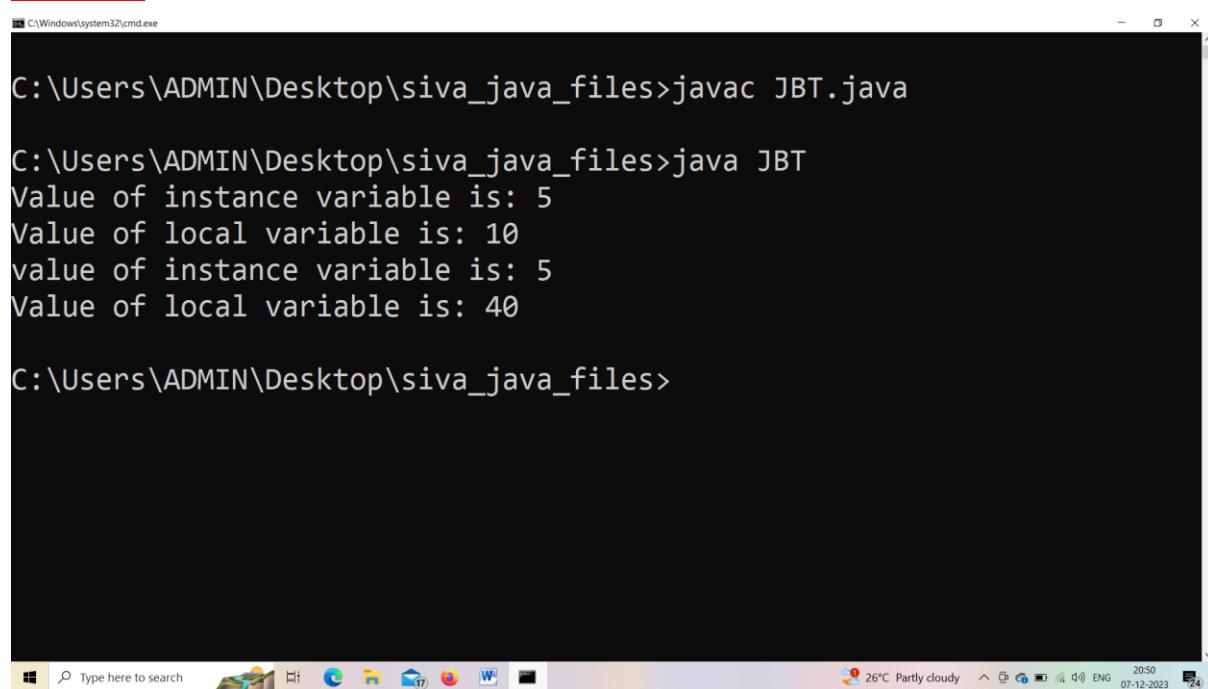
```
C:\Users\ADMIN\Desktop\siva_java_files>javac Without_This.java
C:\Users\ADMIN\Desktop\siva_java_files>java Without_This
0 null
0 null
```

34. Write Another Example Program on THIS keyword.

Program:-

```
class JBT
{
    int variable = 5;
    public static void main(String args[])
    {
        JBT obj = new JBT ();
        obj.method(20);
        obj.method();
    }
    void method (int variable)
    {
        variable = 10;
        System.out.println("Value of instance variable is: "+this.variable);
        System.out.println("Value of local variable is: "+variable);
    }
    void method (){
        int variable = 40;
        System.out.println("value of instance variable is: "+this.variable);
        System.out.println("Value of local variable is: "+variable);
    }
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac JBT.java
C:\Users\ADMIN\Desktop\siva_java_files>java JBT
Value of instance variable is: 5
Value of local variable is: 10
value of instance variable is: 5
Value of local variable is: 40

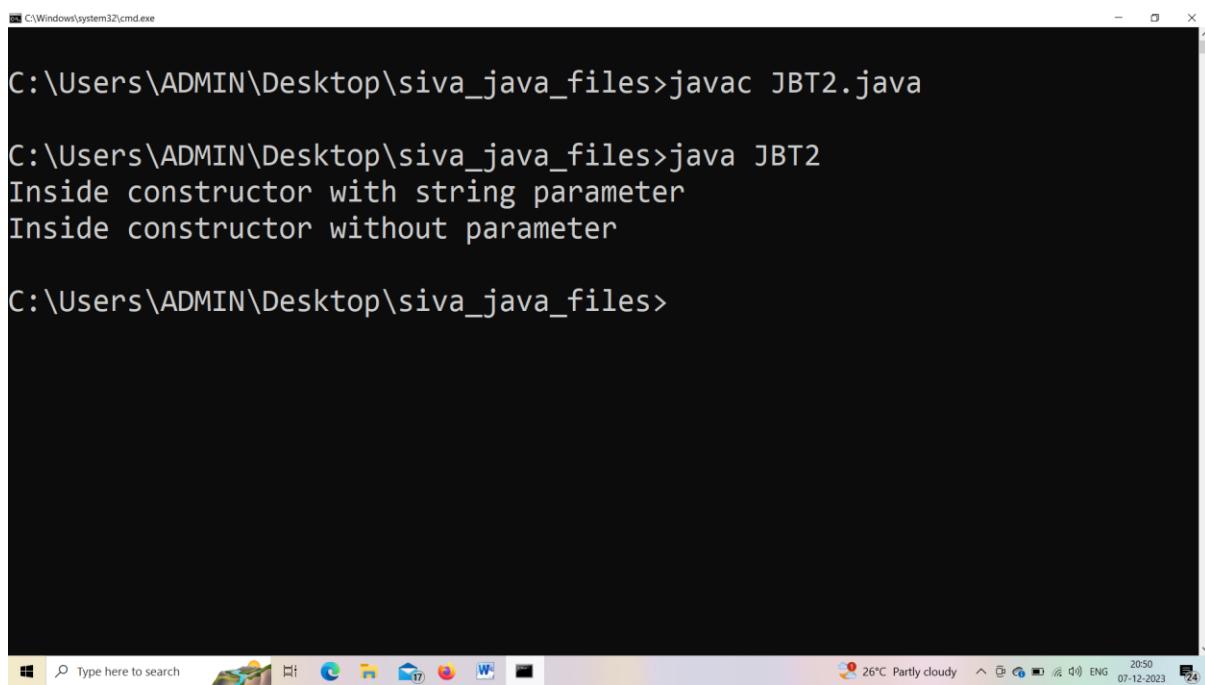
C:\Users\ADMIN\Desktop\siva_java_files>
```

35. Write a Program on usage of THIS keyword in Constructor.

Program:-

```
class JBT2
{
    JBT2()
    {
        this ("JBT2");
        System.out.println("Inside constructor without parameter");
    }
    JBT2(String str)
    {
        System.out.println("Inside constructor with string parameter");
    }
    public static void main (String args[])
    {
        JBT2 obj = new JBT2();
    }
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac JBT2.java
C:\Users\ADMIN\Desktop\siva_java_files>java JBT2
Inside constructor with string parameter
Inside constructor without parameter
C:\Users\ADMIN\Desktop\siva_java_files>
```

36. Write a Program on usage of THIS keyword in method.

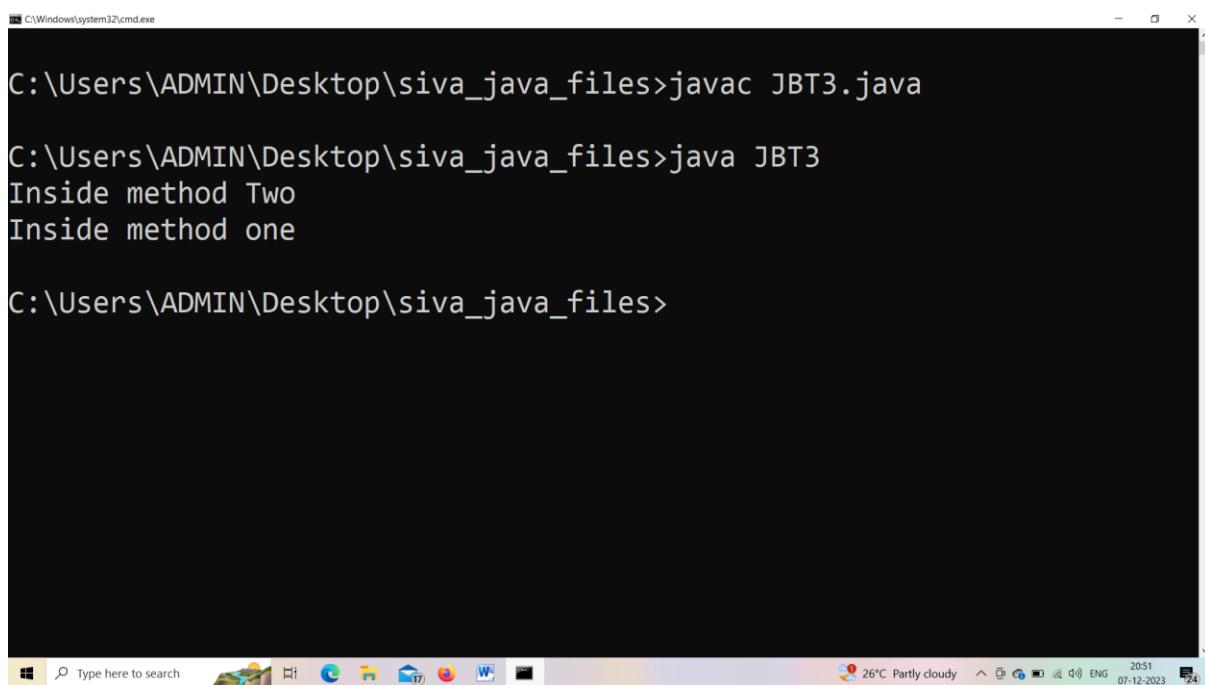
Program:-

```
class JBT3
{
    public static void main(String args[])
    {
        JBT3 obj = new JBT3();
        obj.methodTwo();
    }

    void methodOne()
    {
        System.out.println("Inside method one");
    }

    void methodTwo()
    {
        System.out.println("Inside method Two ");
        this.methodOne();
    }
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac JBT3.java
C:\Users\ADMIN\Desktop\siva_java_files>java JBT3
Inside method Two
Inside method one

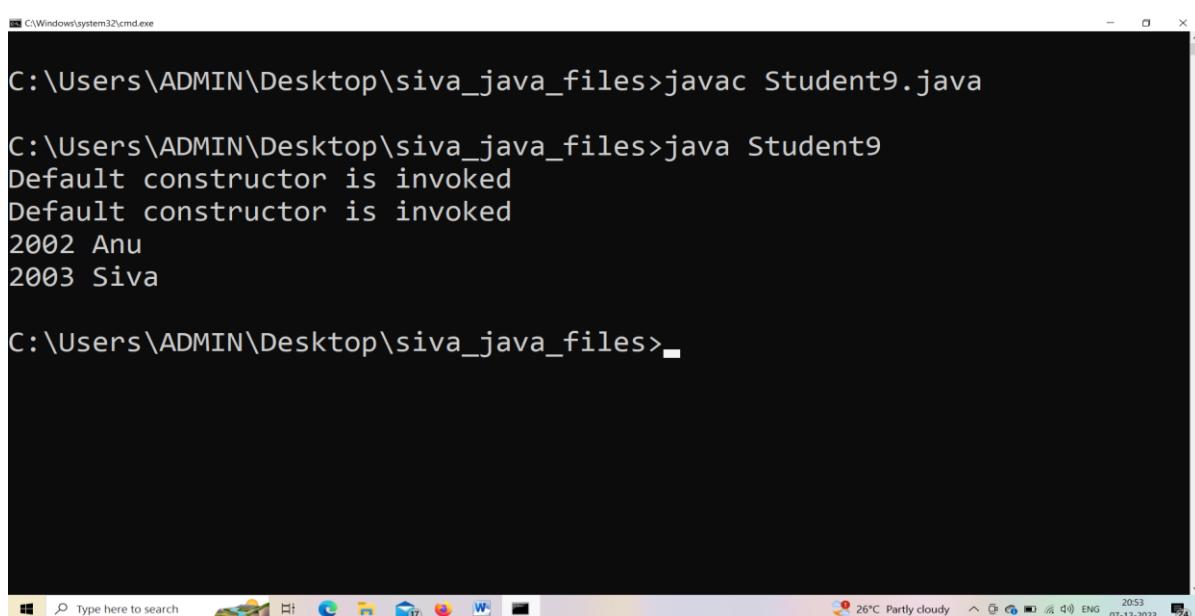
C:\Users\ADMIN\Desktop\siva_java_files>
```

37. Write a Program on THIS() to Invoke Current Class Constructor.

Program:-

```
class Student9 {  
    int id;  
    String name;  
    Student9(){  
        System.out.println("Default constructor is invoked");  
    }  
    Student9(int i, String n){  
        this();  
        id =i;  
        name = n;  
    }  
    void display() {  
        System.out.println(id+" "+name);  
    }  
    public static void main(String args[]){  
        Student9 s1 = new Student9(2002,"Anu");  
        Student9 s2 = new Student9(2003,"Siva");  
        s1.display();  
        s2.display();  
    }  
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac Student9.java' is run, followed by 'java Student9'. The output displays two instances of the 'Student9' class being constructed with default values, and then their display methods are called, outputting '2002 Anu' and '2003 Siva' respectively. The taskbar at the bottom shows various open applications like File Explorer, Edge, and FileZilla.

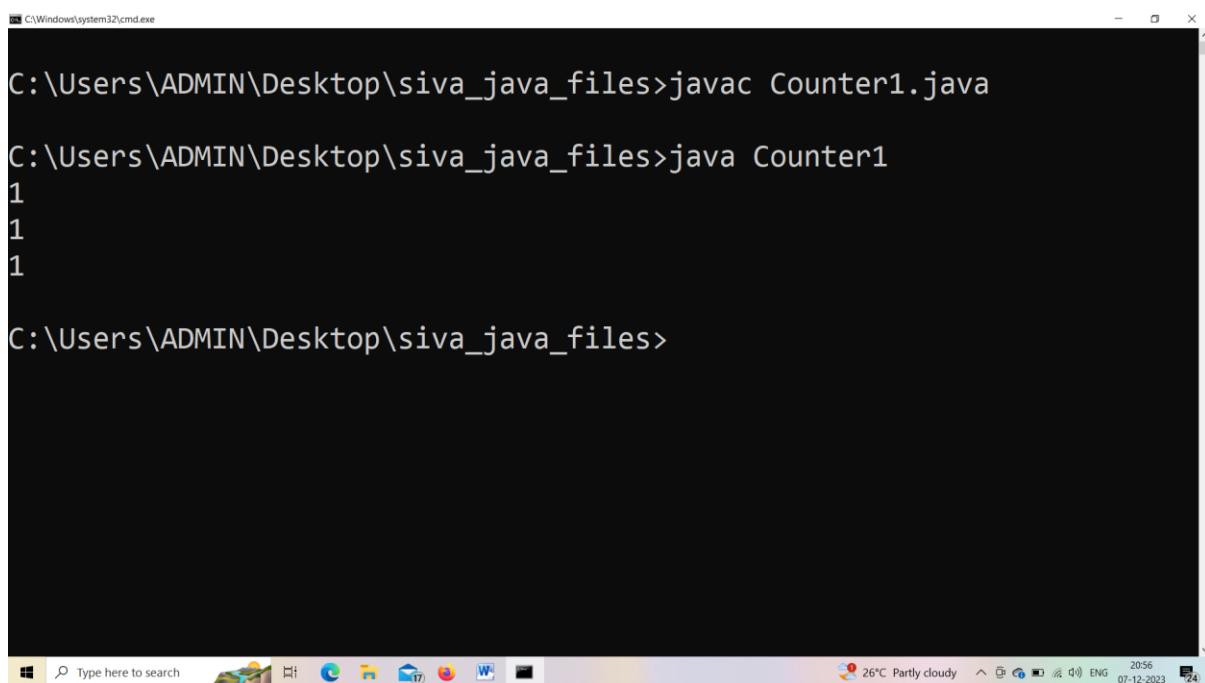
```
C:\Users\ADMIN\Desktop\siva_java_files>javac Student9.java  
C:\Users\ADMIN\Desktop\siva_java_files>java Student9  
Default constructor is invoked  
Default constructor is invoked  
2002 Anu  
2003 Siva
```

38. Write a Program on Counter Variable without Static keyword.

Program:-

```
class Counter1
{
int count=0;
Counter1()
{
count++;
System.out.println(count);
}
public static void main(String args[])
{
Counter1 c1 = new Counter1();
Counter1 c2 = new Counter1();
Counter1 c3 = new Counter1();
}
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac Counter1.java' is run, followed by 'java Counter1'. The output shows three '1's printed to the console, indicating that each Counter1 object has its own separate counter variable. The taskbar at the bottom includes icons for File Explorer, Edge, Mail, and File Explorer, along with system status indicators like battery level and network connection.

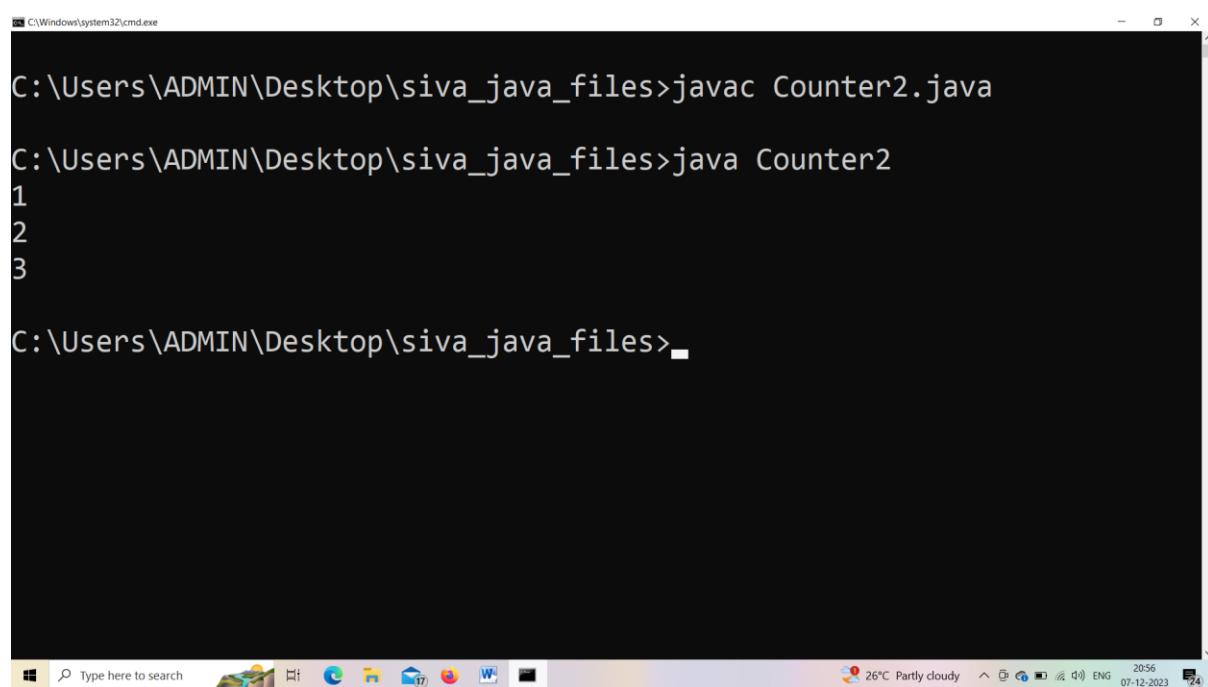
```
C:\Users\ADMIN\Desktop\siva_java_files>javac Counter1.java
C:\Users\ADMIN\Desktop\siva_java_files>java Counter1
1
1
1
```

39. Write a Program on Counter Variable with Static keyword.

Program:-

```
class Counter2
{
    static int count=0;
    Counter2()
    {
        count++;
        System.out.println(count);
    }
    public static void main (String args[])
    {
        Counter2 c1 = new Counter2();
        Counter2 c2 = new Counter2();
        Counter2 c3 = new Counter2();
    }
}
```

Output:-



```
C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac Counter2.java
C:\Users\ADMIN\Desktop\siva_java_files>java Counter2
1
2
3

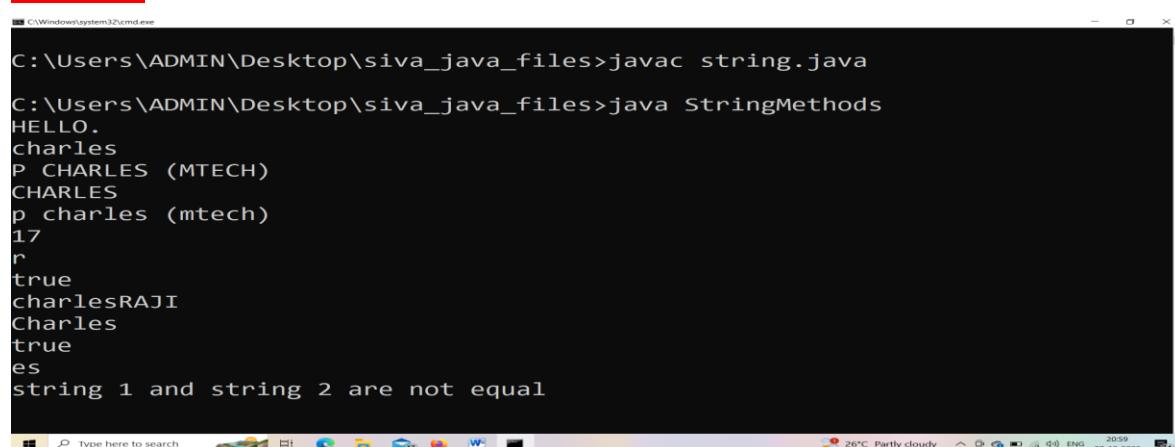
C:\Users\ADMIN\Desktop\siva_java_files>
```

40. Write a program to implement String Methods.

Program:-

```
class StringMethods
{
    public static void main(String args[])
    {
        String s1=new String("charles");
        String s2="P "+"CHARLES "+"(MTECH)";
        String s3=s1;
        char[] helloArray={'H','E','L','L','O','.'};
        String s4=new String(helloArray);
        System.out.println(s4);
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s1.toUpperCase());
        System.out.println(s2.toLowerCase());
        System.out.println(s2.length());
        System.out.println(s1.charAt(3));
        System.out.println(s1.equals(s3));
        System.out.println(s1.concat("RAJI"));
        System.out.println(s1.replace('c','C'));
        System.out.println(s1.equalsIgnoreCase("CHARLES"));
        System.out.println(s1.substring(5));
        if(s1.equals(s2))
            System.out.println("string 1 and string 2 are equal");
        else
            System.out.println("string 1 and string 2 are not equal");
    }
}
```

Output:-



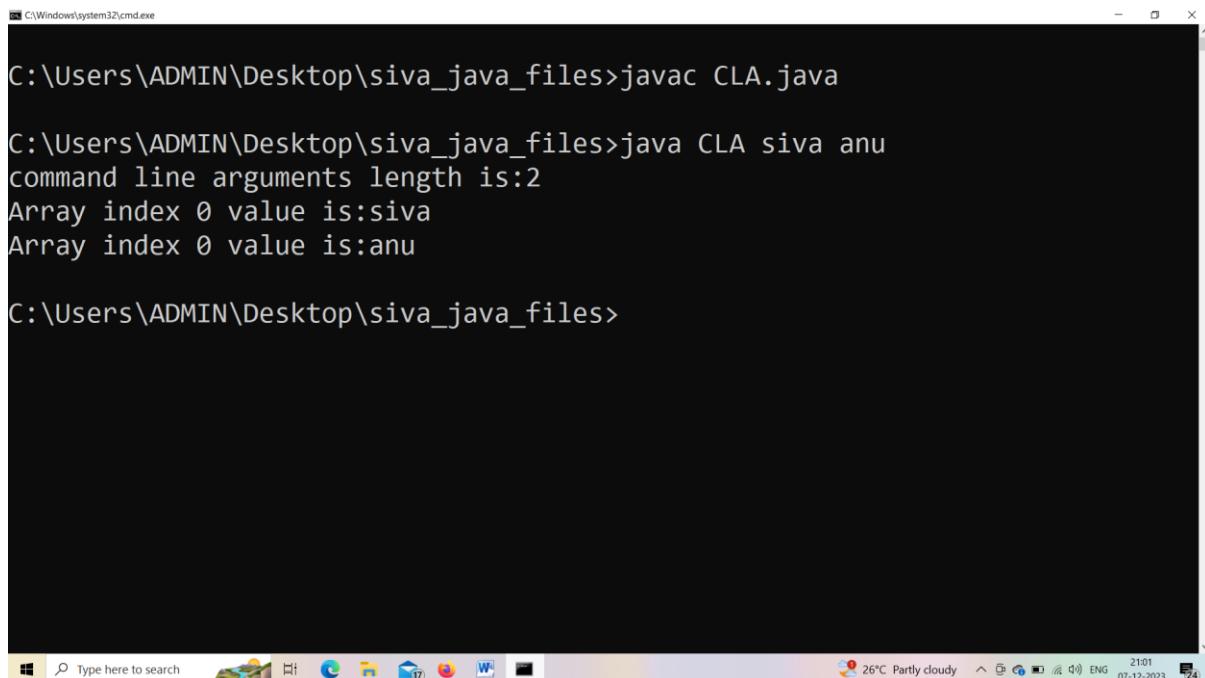
```
C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac string.java
C:\Users\ADMIN\Desktop\siva_java_files>java StringMethods
HELLO.
charles
P CHARLES (MTECH)
CHARLES
p charles (mtech)
17
r
true
charlesRAJI
Charles
true
es
string 1 and string 2 are not equal
```

41. Write a program on Command Line Arguments

Program:-

```
class CLA
{
    public static void main(String args[])
    {
        System.out.println("command line arguments length is:"+args.length);
        System.out.println("Array index 0 value is:"+args[0]);
        System.out.println("Array index 0 value is:"+args[1]);
    }
}
```

Output:-



```
C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac CLA.java
C:\Users\ADMIN\Desktop\siva_java_files>java CLA siva anu
command line arguments length is:2
Array index 0 value is:siva
Array index 0 value is:anu

C:\Users\ADMIN\Desktop\siva_java_files>
```

42. write a program to implement hybrid inheritance.

Program:-

```
class A
{
Public void methodA()
{
System.out.println("class A method A");
}
}

class B extends A
{
public void methodA()
{
System.out.println("child class B is over ridding inherited method");
}
public void methodB()
{
System.out.println("class B method B");
}
}

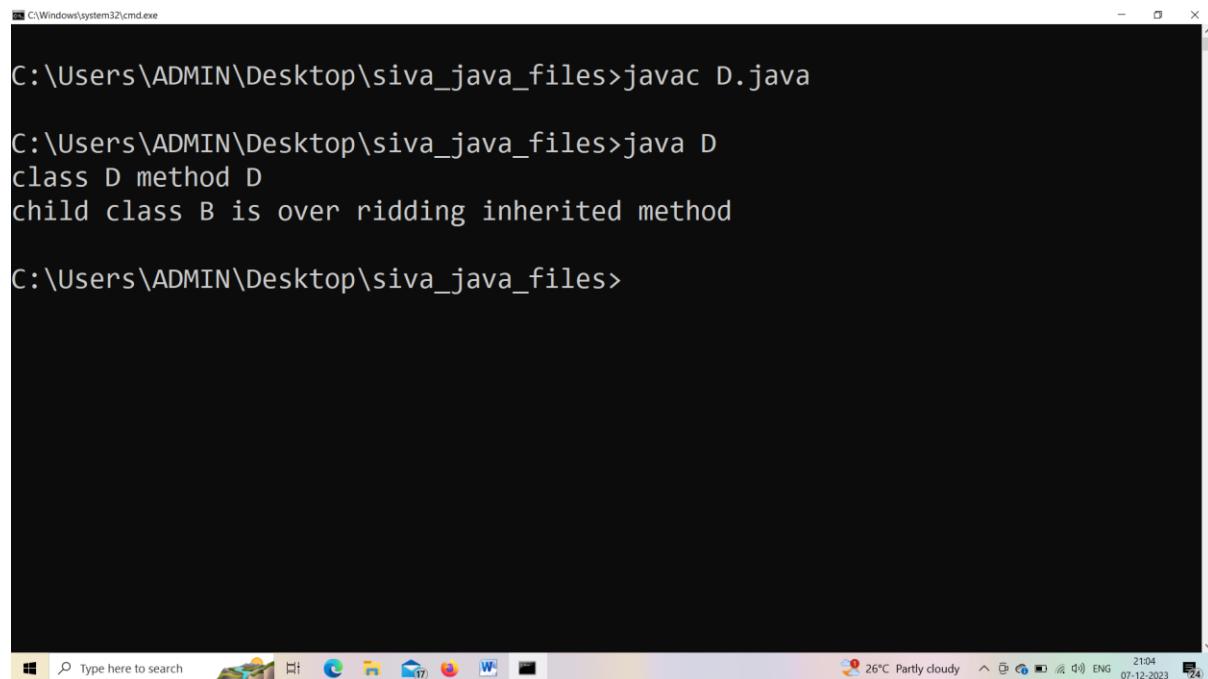
class C extends A
{
public void methodA()
{
System.out.println("child class C is over ridding inherited method");
}
public void methodC()
{
System.out.println("class C method C");
}
}

class D extends B
{
public void methodD()
{
System.out.println("class D method D");
}
}

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

```
{  
D obj1=new D();  
obj1.methodD();  
obj1.methodA();  
}  
}
```

Output:-



C:\Windows\system32\cmd.exe

```
C:\Users\ADMIN\Desktop\siva_java_files>javac D.java  
C:\Users\ADMIN\Desktop\siva_java_files>java D  
class D method D  
child class B is overriding inherited method  
C:\Users\ADMIN\Desktop\siva_java_files>
```

The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\cmd.exe". The user has run the command "javac D.java" followed by "java D". The output indicates that the class D contains a method D, and a child class B is overriding the inherited method D. The Java environment includes standard icons for file operations and system status at the bottom.

43. Write a Program to implement Method Overriding with Super.

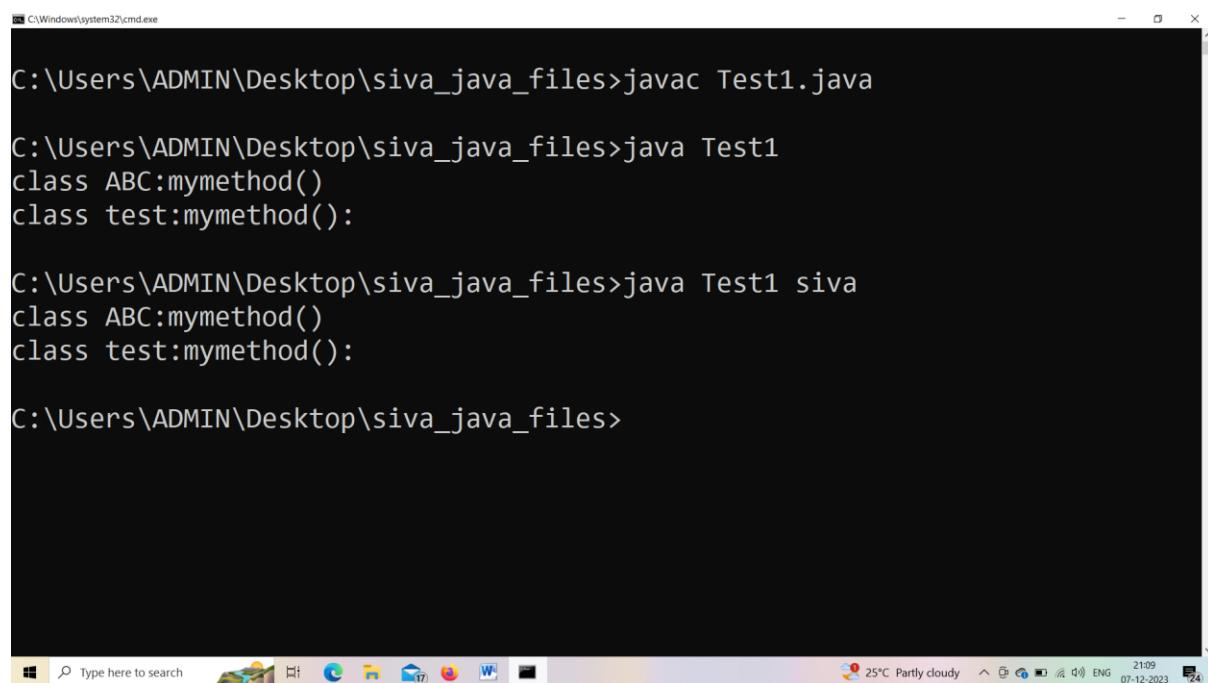
Program:-

```
class ABC {
    public void mymethod()
    {
        System.out.println("class ABC:mymethod()");
    }
}

class Test1 extends ABC
{
    public void mymethod()
    {
        super.mymethod();
        System.out.println("class test:mymethod()");
    }

    public static void main(String args[])
    {
        Test1 obj = new Test1();
        obj.mymethod();
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac Test1.java' is run, followed by 'java Test1'. The output shows the method signature from the ABC class and then the overridden method from the Test1 class. The taskbar at the bottom includes icons for File Explorer, Edge, Mail, and File History, along with system status indicators like battery level and network connection.

```
C:\Users\ADMIN\Desktop\siva_java_files>javac Test1.java
C:\Users\ADMIN\Desktop\siva_java_files>java Test1
class ABC:mymethod()
class test:mymethod():

C:\Users\ADMIN\Desktop\siva_java_files>java Test1 siva
class ABC:mymethod()
class test:mymethod():

C:\Users\ADMIN\Desktop\siva_java_files>
```

44. Write a demo program on creating packages.

Program:-

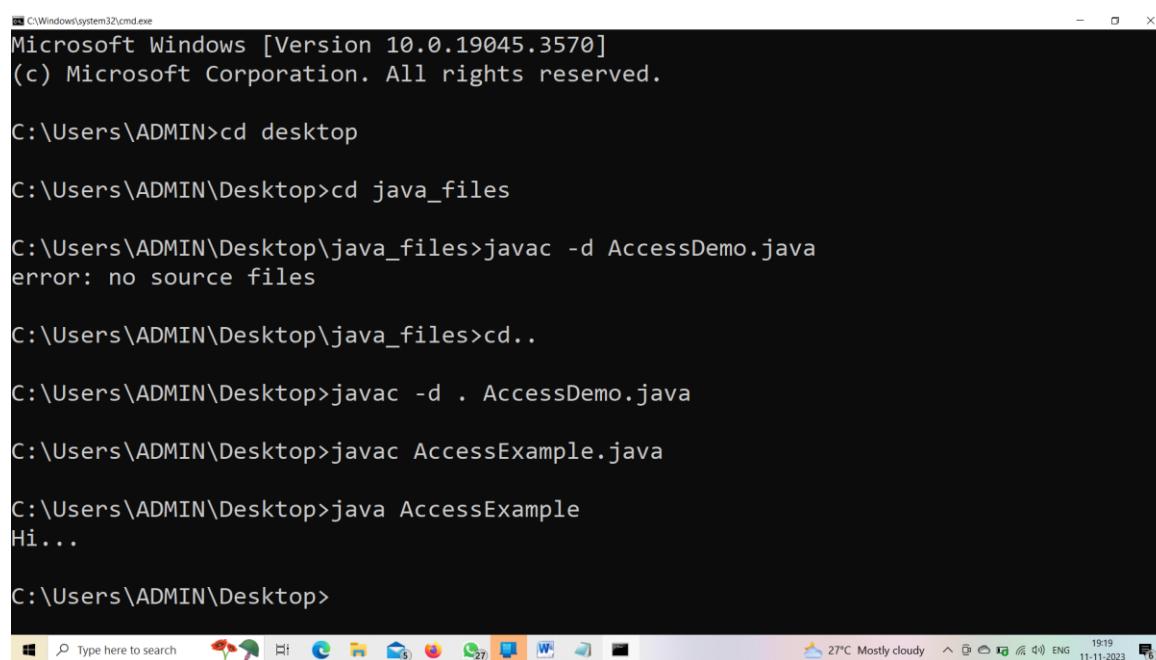
AccessDemo.java:-

```
package abc;
public class AccessDemo
{
    public void test()
    {
        System.out.println("Hi... ");
    }
}
```

AccessExample.java:-

```
import abc.AccessDemo;
public class AccessExample
{
    public static void main(String args[])
    {
        AccessDemo ad=new AccessDemo();
        ad.test();
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The window displays the following command-line session:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>cd desktop
C:\Users\ADMIN\Desktop>cd java_files
C:\Users\ADMIN\Desktop\java_files>javac -d AccessDemo.java
error: no source files

C:\Users\ADMIN\Desktop\java_files>cd..
C:\Users\ADMIN\Desktop>javac -d . AccessDemo.java
C:\Users\ADMIN\Desktop>javac AccessExample.java
C:\Users\ADMIN\Desktop>java AccessExample
Hi...

C:\Users\ADMIN\Desktop>
```

The taskbar at the bottom of the screen includes icons for File Explorer, Edge browser, Mail, Task View, Word, and Powerpoint. The system tray shows the date (11-11-2023), time (19:19), battery level (6%), and network status.

45. Write a Program on Interface.

Program:-

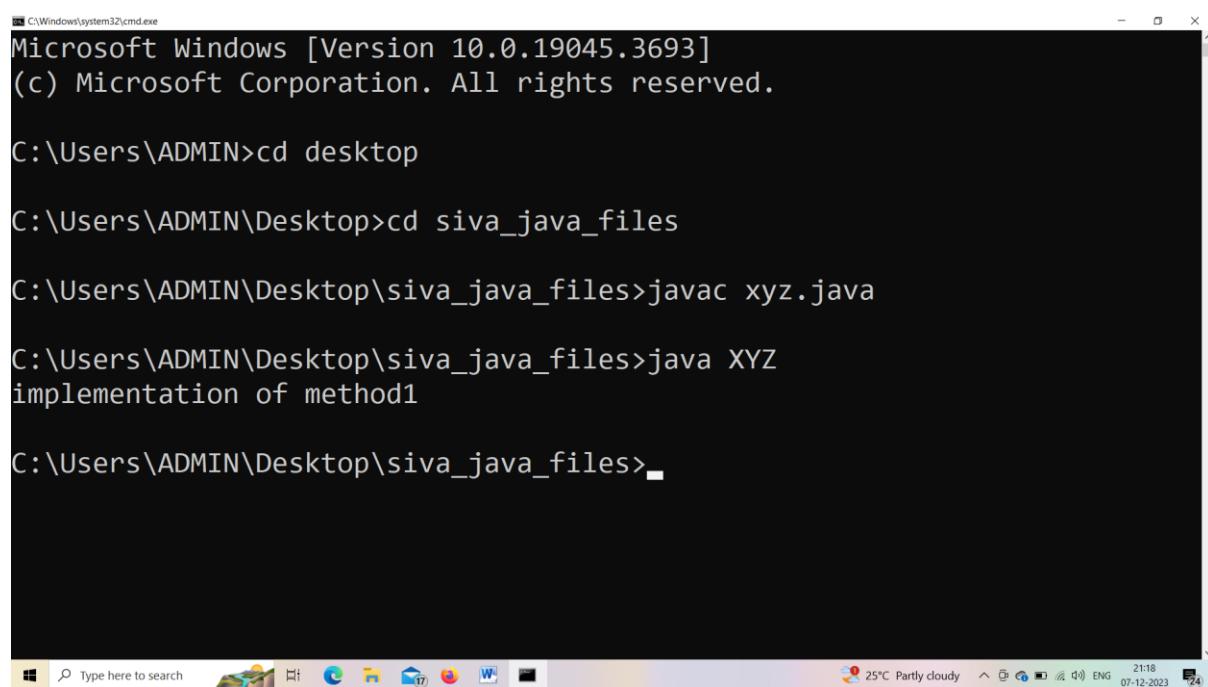
```
interface Myinterface
{
    public void method1();
    public void method2();
}

class XYZ implements Myinterface
{
    public void method1()
    {
        System.out.println("implementation of method1");
    }

    public void method2()
    {
        System.out.println("implementation of method2");
    }

    public static void main(String args[])
    {
        Myinterface obj = new XYZ();
        obj.method1();
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The window displays the following command-line session:

```
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>cd desktop

C:\Users\ADMIN\Desktop>cd siva_java_files

C:\Users\ADMIN\Desktop\siva_java_files>javac xyz.java

C:\Users\ADMIN\Desktop\siva_java_files>java XYZ
implementation of method1

C:\Users\ADMIN\Desktop\siva_java_files>
```

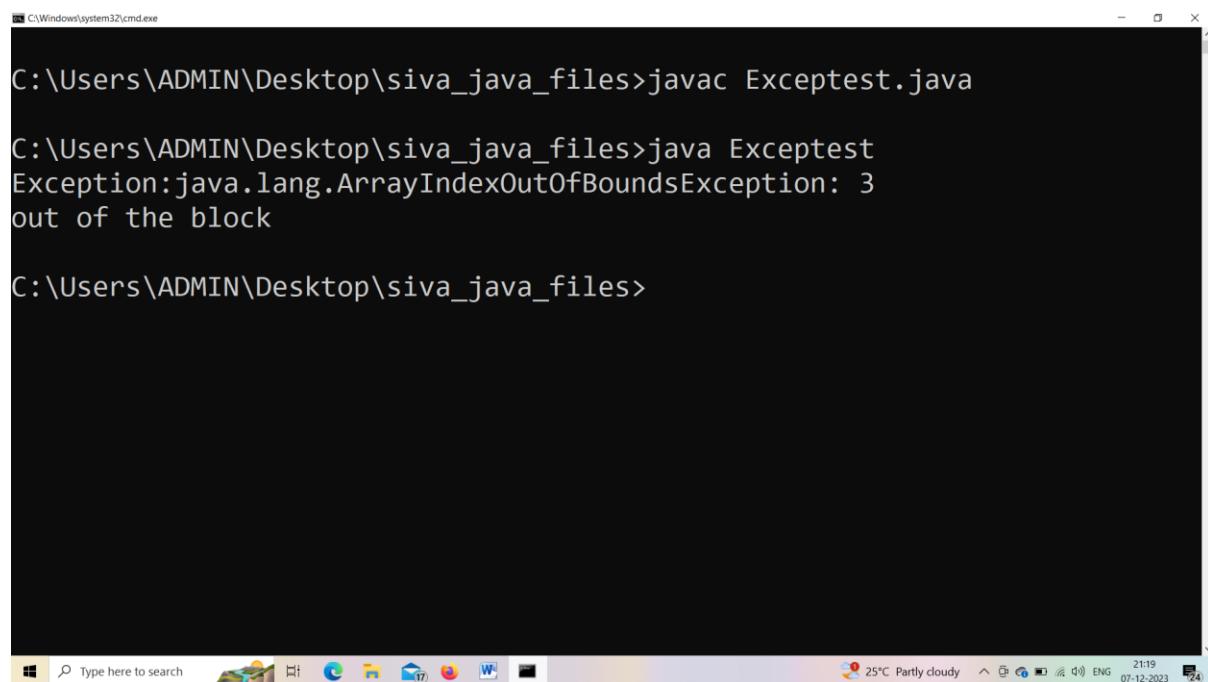
The taskbar at the bottom of the screen includes icons for File Explorer, Edge browser, Mail, and File Explorer again. The system tray shows the date and time as '07-12-2023' and '21:18'.

46. Write a program to implement try, catch blocks.

Program:-

```
import java.io.*;
public class Exceptest
{
    public static void main(String args[])
    {
        try
        {
            int a[] = new int[2];
            System.out.println("accessing element three:"+a[3]);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Exception:"+e);
        }
        System.out.println("out of the block");
    }
}
```

Output:-



```
C:\Users\ADMIN\Desktop\siva_java_files>javac Exceptest.java
C:\Users\ADMIN\Desktop\siva_java_files>java Exceptest
Exception:java.lang.ArrayIndexOutOfBoundsException: 3
out of the block

C:\Users\ADMIN\Desktop\siva_java_files>
```

47. Write a program to implement Exception Handling Mechanisms of try, catch and finally blocks.

Program:-

```
public class Myfinallyblock
{
    public static void main(String args[])
    {
        try
        {
            int i=10/0;
        }
        catch (ArithmaticException ex)
        {
            System.out.println("inside 1st catch block");
        }
        finally
        {
            System.out.println("inside 1st finally block");
        }
        try
        {
            int i=10/10;
        }
        catch(ArithmaticException ex)
        {
            System.out.println("inside 2nd catch block");
        }
        finally
        {
            System.out.println("inside 2nd finally block");
        }
    }
}
```

Output:-

```
C:\Windows\system32\cmd.exe

C:\Users\ADMIN\Desktop\siva_java_files>javac Myfinallyblock.java

C:\Users\ADMIN\Desktop\siva_java_files>java Myfinallyblock
inside 1st catch block
inside 1st finally block
inside 2nd finally block

C:\Users\ADMIN\Desktop\siva_java_files>
```

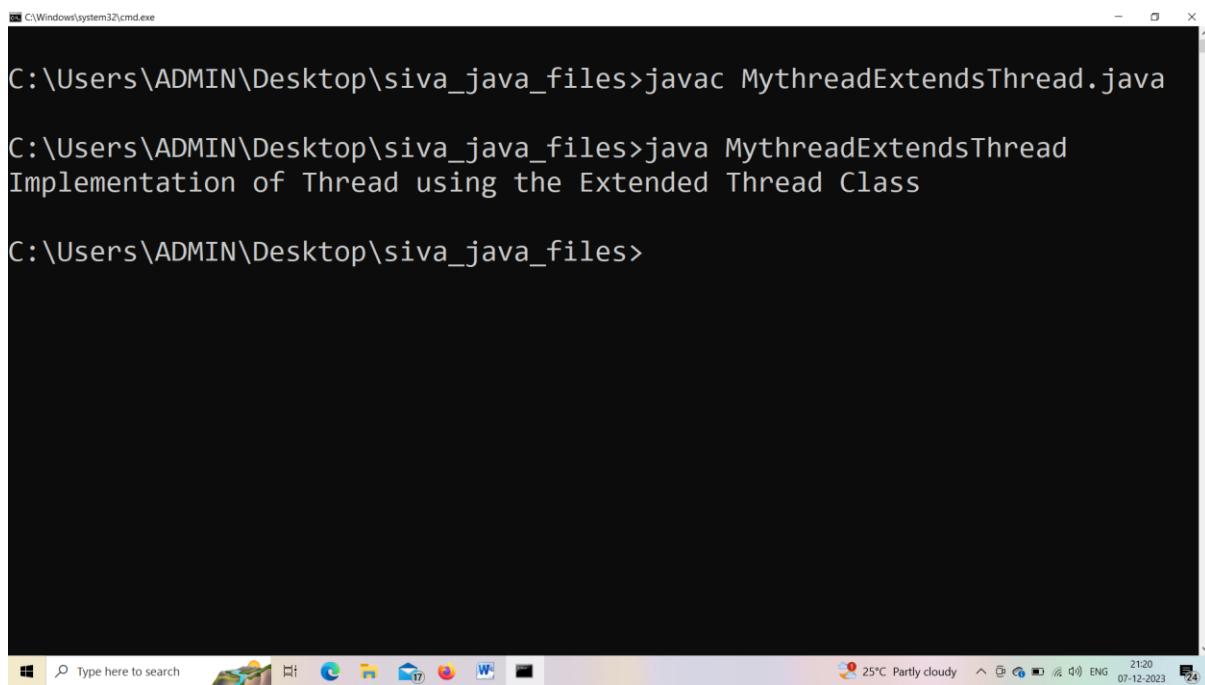
48. Write a program to create our own Thread using Thread Class.

Program:-

```
Class Mythread extends Thread
{
    public void run()
    {
        System.out.println("Implementation of Thread using the Extended Thread Class");
    }
}

Class MythreadExtendsThread
{
    public static void main(String args[])
    {
        Mythread mt = new Mythread();
        mt.start();
    }
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The command 'javac MythreadExtendsThread.java' is entered and executed, followed by the command 'java MythreadExtendsThread'. The output of the Java application is displayed, showing the message 'Implementation of Thread using the Extended Thread Class'. The taskbar at the bottom of the screen includes icons for File Explorer, Edge browser, Mail, and File Explorer, along with system status indicators like weather (25°C Partly cloudy), date (07-12-2023), and battery level (24%).

```
C:\Users\ADMIN\Desktop\siva_java_files>javac MythreadExtendsThread.java
C:\Users\ADMIN\Desktop\siva_java_files>java MythreadExtendsThread
Implementation of Thread using the Extended Thread Class
C:\Users\ADMIN\Desktop\siva_java_files>
```

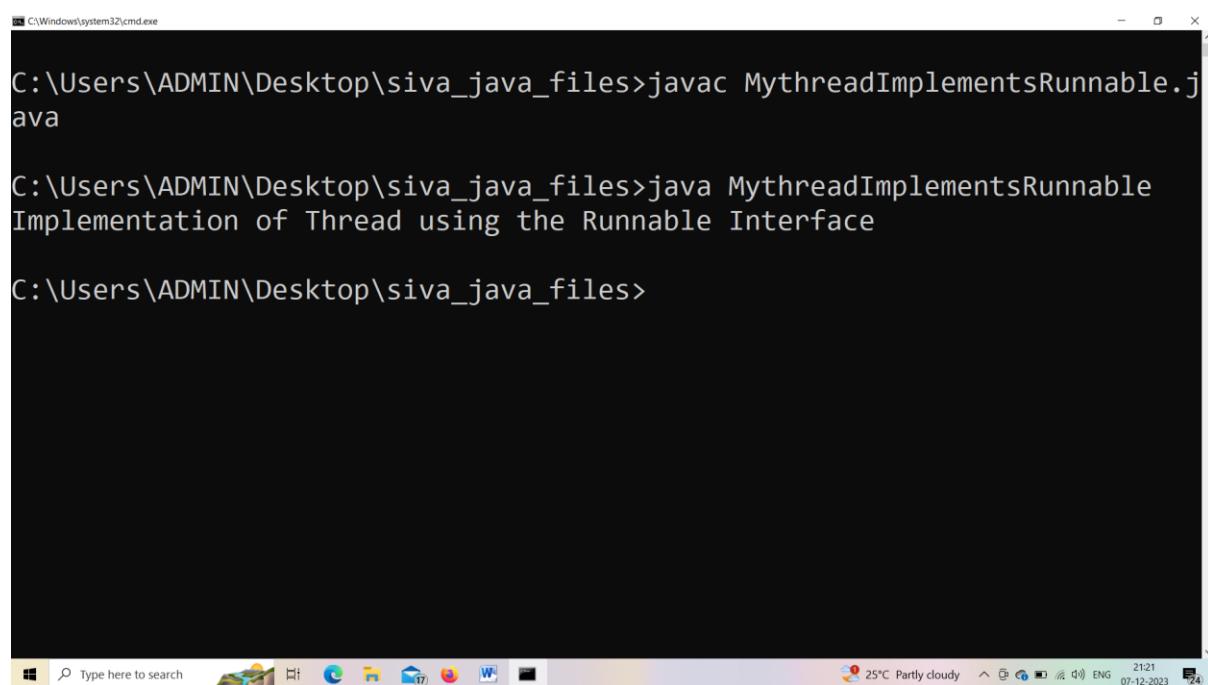
49. Write a program to create our own Thread using Runnable Interface (Implements).

Program:-

```
class Mymethod implements Runnable
{
public void run()
{
System.out.println("Implementation of Thread using the Runnable Interface");
}
}

class MythreadImplementsRunnable
{
public static void main(String args[])
{
Runnable r=new Mymethod();
Thread t = new Thread(r);
t.start();
}
}
```

Output:-



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The command 'javac MythreadImplementsRunnable.java' is entered and executed, followed by 'java MythreadImplementsRunnable', which prints the message 'Implementation of Thread using the Runnable Interface' to the console. The taskbar at the bottom shows various application icons, and the system tray indicates the date as 07-12-2023.

```
C:\Users\ADMIN\Desktop\siva_java_files>javac MythreadImplementsRunnable.java
C:\Users\ADMIN\Desktop\siva_java_files>java MythreadImplementsRunnable
Implementation of Thread using the Runnable Interface
C:\Users\ADMIN\Desktop\siva_java_files>
```

50. Write a Program to Implement Thread Methods using Thread Class.

Program:-

```
class MyThread extends Thread
{
public void run()
{
for(int i=0;i<=5;i++)
{
try
{
Thread.sleep(1000);
}
catch(Exception e)
{
}
System.out.println(i);
}
}

class ThreadMethods
{
public static void main(String args[])
{
MyThread mt = new MyThread();
System.out.println("Thread name is:"+mt.getName());
mt.setName("Siva");
System.out.println("Thread name after setting new name is:"+mt.getName());
System.out.println("Thread Priority is:"+mt.getPriority());
mt.setPriority(1);
System.out.println("After Changing Thread Priority is:"+mt.getPriority());
System.out.println("Thread ID is:"+mt.getId());
mt.start();
}
}
```

Output:-

```
C:\Windows\system32\cmd.exe
C:\Users\ADMIN\Desktop\siva_java_files>javac ThreadMethods.java
C:\Users\ADMIN\Desktop\siva_java_files>java ThreadMethods
Thread name is:Thread-0
Thread name after setting new name is:Siva
Thread Priority is:5
After Changing Thread Priority is:1
Thread ID is:10
0
1
2
3
4
5
C:\Users\ADMIN\Desktop\siva_java_files>
```

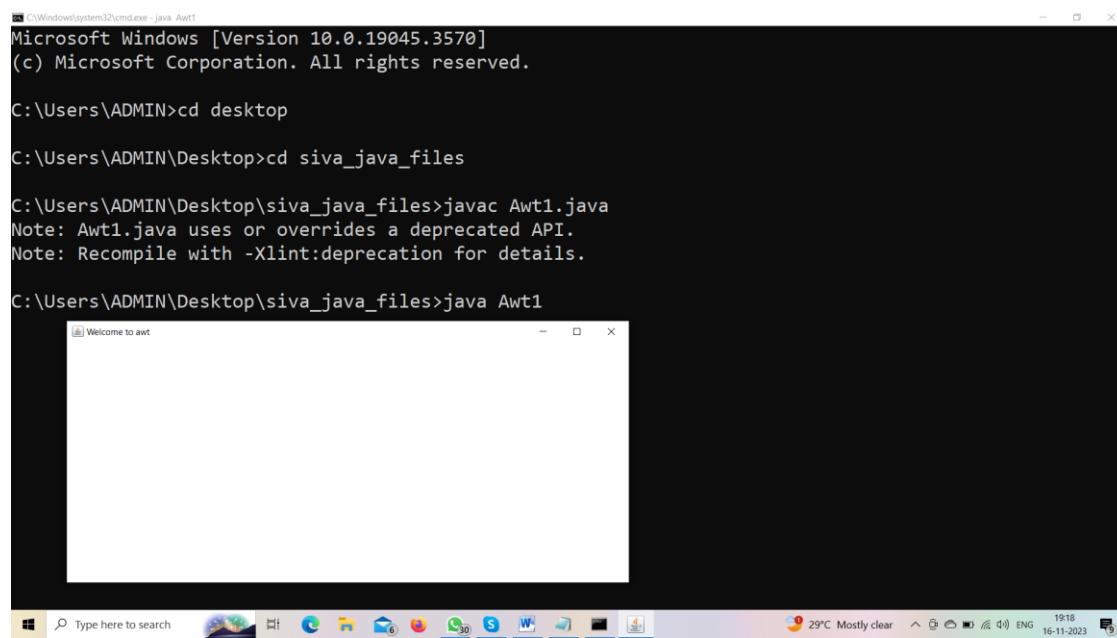
JAVA FRAMES

Program:- 1

Java program to create frame by frame class

```
import java.awt.*;
class Awt1
{
Awt1(String s)
{
Frame f = new Frame();
f.setTitle(s);
f.setSize(200,200);
f.show();
}
public static void main(String[] args)
{
Awt1 a = new Awt1("Welcome to awt");
}
```

Output:-

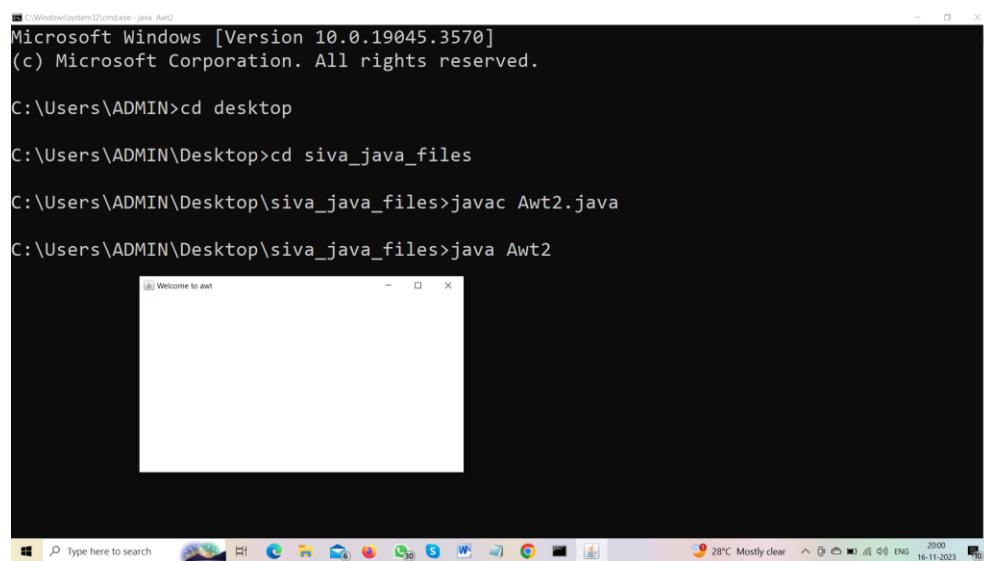


Program:- 2

Creating frame by extending frame class

```
import java.awt.*;
class Awt2 extends Frame
{
Awt2(String s)
{
super(s);
setSize(200,200);
setVisible(true);
}
public static void main(String[] args)
{
Awt2 a = new Awt2("Welcome to awt");
}
}
```

Output:-

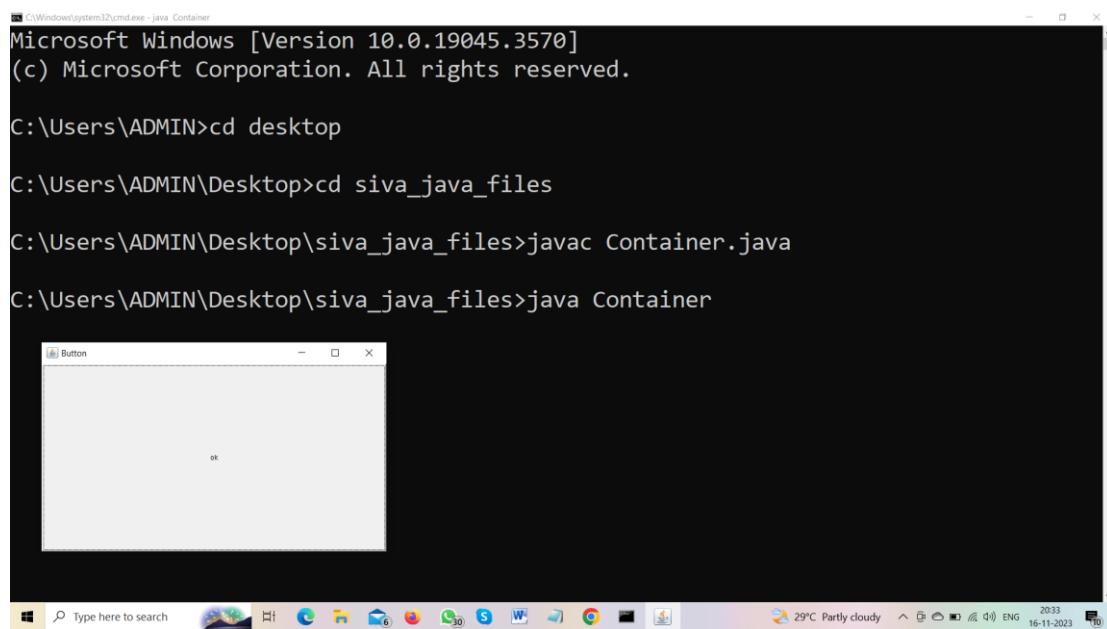


Program:- 3

Java program to create a button

```
import java.awt.*;
class Container extends Frame
{
Container(String s)
{
super(s);
setSize(200,200);
setVisible(true);
Button b = new Button ("ok");
add(b);
}
public static void main(String[] args)
{
Container b = new Container ("Button");
}
```

Output:-

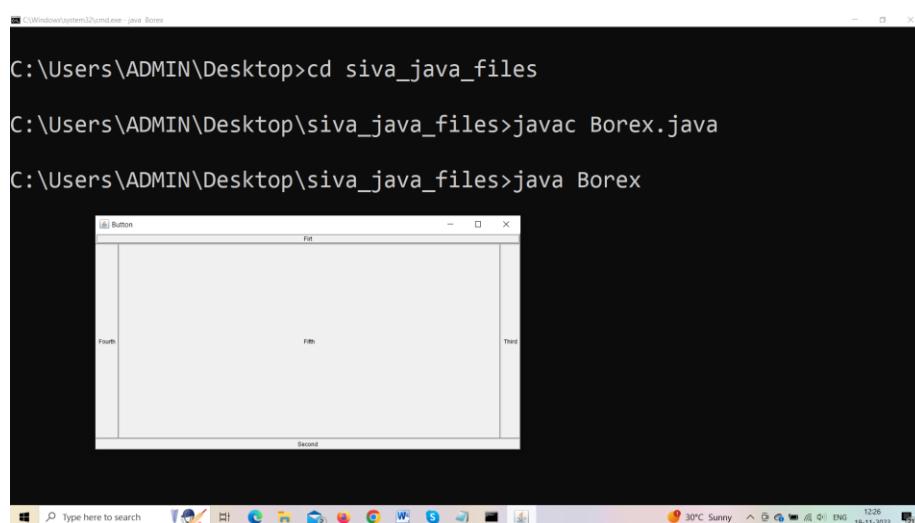


Program:- 4

Java program to create a BorderLayout

```
import java.awt.*;
class Borex extends Frame
{
Borex(String s)
{
super(s);
Button b1,b2,b3,b4,b5;
b1= new Button ("Firt");
b2= new Button ("Second");
b3= new Button ("Third");
b4= new Button ("Fourth");
b5= new Button ("Fifth");
add(b1, "North");
add(b2, "South");
add(b3, "East");
add(b4, "West");
add(b5, "Center");
}
public static void main(String[] args)
{
Borex b= new Borex("Button");
b.setSize(500,500);
b.setVisible(true);
}
```

Output:-

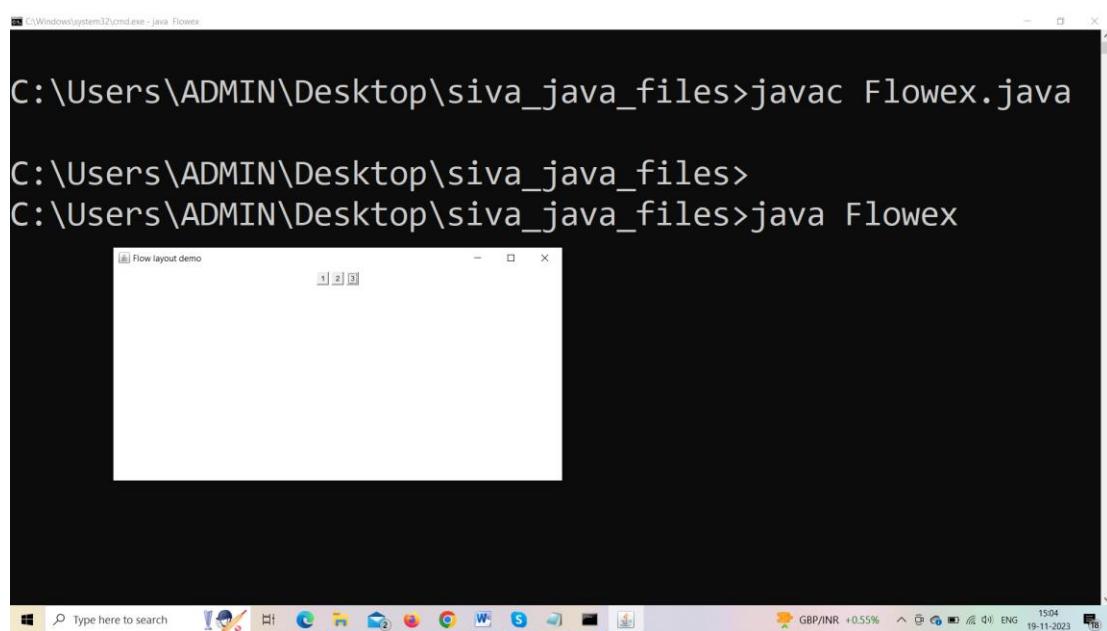


Program:- 5

Write a Java program to create a Flow layout

```
import java.awt.*;
class Flowex extends Frame
{
Flowex(String tit)
{
super(tit);
Button b1,b2,b3;
b1= new Button("1");
b2= new Button("2");
b3= new Button("3");
setLayout(new FlowLayout());
add(b1);
add(b2);
add(b3);
}
public static void main(String[] args)
{
Flowex f1 = new Flowex("Flow layout demo");
f1.setSize(400,400);
f1.setVisible(true);
}
```

Output:-

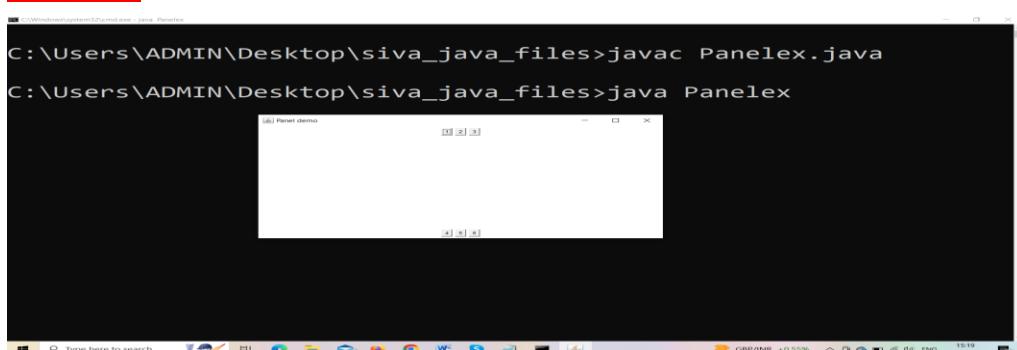


Program:- 6

Java program to create a Panel Layout

```
import java.awt.*;
class Panelex extends Frame
{
Panelex(String tit)
{
super (tit);
Button b1,b2,b3,b4,b5,b6;
b1=new Button("1");
b2=new Button("2");
b3=new Button("3");
b4=new Button("4");
b5=new Button("5");
b6=new Button("6");
Panel p1= new Panel();
p1.add(b1);
p1.add(b2);
p1.add(b3);
add(p1,BorderLayout.NORTH);
Panel p2 = new Panel();
p2.add(b4);
p2.add(b5);
p2.add(b6);
add(p2,BorderLayout.SOUTH);
}
public static void main(String[] args)
{
Panelex fl = new Panelex("Panel demo");
fl.setSize(400,400);
fl.setVisible(true);
}
```

Output:-



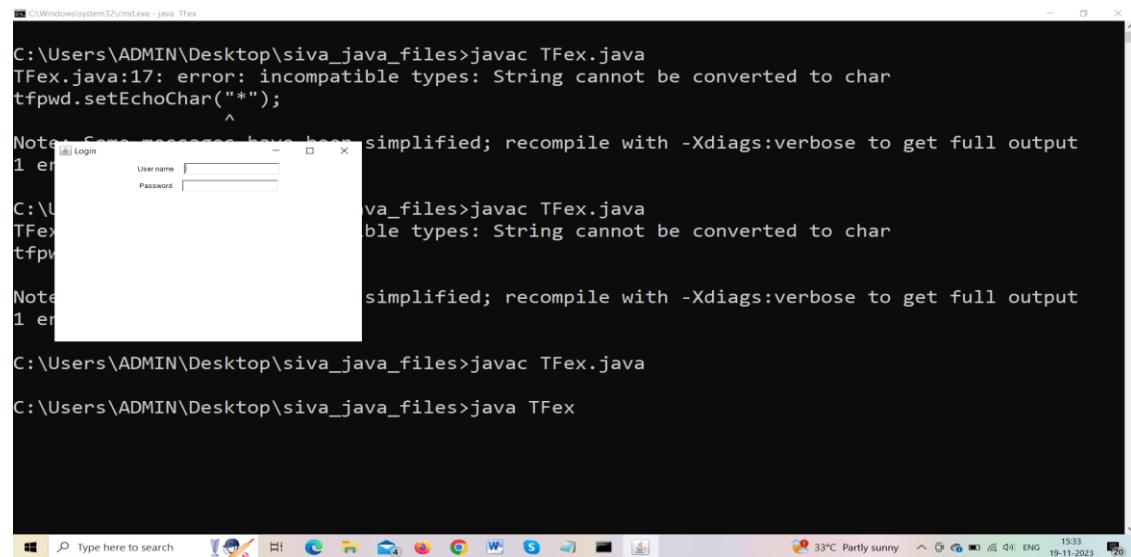
Program:- 7

Java program to create a Textfield and Label

```
import java.awt.*;  
  
class TFex extends Frame  
  
{  
  
    TFex(String tit)  
  
    {  
        super(tit);  
  
        Label luser,lpwd;  
  
        TextField tfuser,tfpwd;  
  
        luser = new Label("User name");  
  
        tfuser = new TextField(20);  
  
        Panel p1 = new Panel();  
  
        p1.add(luser);  
  
        p1.add(tfuser);  
  
        add(p1,"North");  
  
        lpwd = new Label("Password");  
  
        tfpwd = new TextField(20);  
  
        tfpwd.setEchoChar('*');  
  
        Panel p2 = new Panel();  
  
        p2.add(lpwd);  
  
        p2.add(tpwd);  
  
        add(p2,"Center");  
    }  
  
    public static void main(String[] args)  
    {  
        TFex fl = new TFex("Login");  
  
        fl.setSize(400,400);
```

```
fl.setVisible(true);  
}  
}  
}
```

Output:-

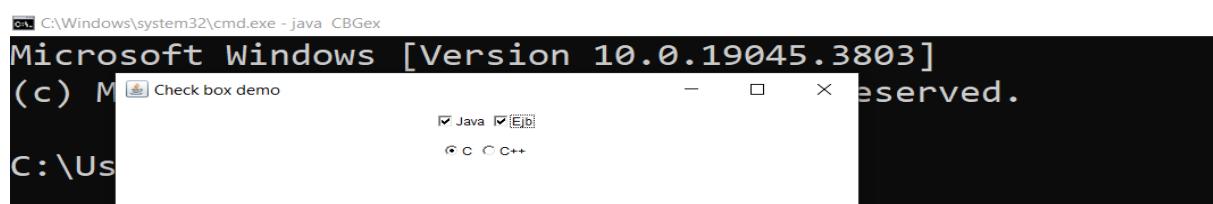


Program:- 8

Java program on Checkbox

```
import java.awt.*;
class CBGex extends Frame
{
    Checkbox cbjava, cbejb, cbc, cbcpp;
    CBGex (String tit)
    {
        super(tit);
        cbjava = new Checkbox("Java");
        cbejb = new Checkbox("Ejb");
        Panel p1 = new Panel();
        p1.add(cbjava);
        p1.add(cbejb);
        add(p1, BorderLayout.NORTH);
        CheckboxGroup cg = new CheckboxGroup();
        cbc = new Checkbox("C",true,cg);
        cbcpp = new Checkbox("C++",false,cg);
        Panel p2 = new Panel();
        p2.add(cbc);
        p2.add(cbcpp);
        add(p2,BorderLayout.CENTER);
    }
    public static void main(String[] args)
    {
        CBGex fl = new CBGex("Check box demo");
        fl.setSize(400,400);
        fl.setVisible(true);
    }
}
```

Output:-



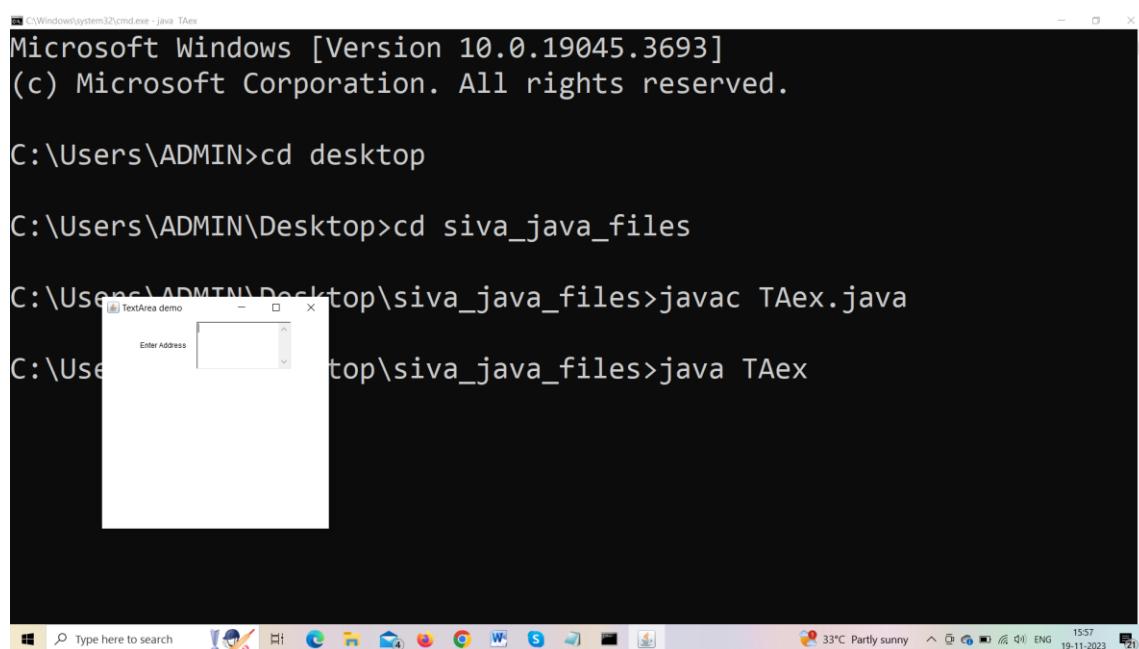
```
C:\Windows\system32\cmd.exe - java CBGex
Microsoft Windows [Version 10.0.19045.3803]
(c) M Check box demo - □ × Reserved.
C:\Us
C:\Users\ADMIN\Desktop>cd siva_java_files
C:\Users\ADMIN\Desktop\siva_java_files>javac CBGex.java
C:\Users\ADMIN\Desktop\siva_java_files>java CBGex
```

Program:- 9

Java program on create TextArea

```
import java.awt.*;
class TAex extends Frame
{
TAex(String title)
{
super (title);
Label L3 = new Label("Enter Address");
TextArea ta = new TextArea(4,20);
Panel p1 = new Panel();
p1.add(L3);
p1.add(ta);
add(p1,BorderLayout.CENTER);
}
public static void main(String[] args)
{
TAex fl = new TAex("TextArea demo");
fl.setSize(400,400);
fl.setVisible(true);
}
}
```

Output:-



Program:- 10

Java program to create a Grid Layout

```
import java.awt.*;
class Gridex extends Frame
{
Gridex(String tit)
{
super(tit);
Button b1,b2,b3,b4;
TextField t1,t2,t3,t4;
b1 = new Button("1");
b2 = new Button("2");
b3 = new Button("3");
b4 = new Button("4");
t1 = new TextField(" ");
t2 = new TextField(" ");
t3 = new TextField(" ");
t4 = new TextField(" ");
setLayout(new GridLayout(2,4));
add(b1);
add(b2);
add(b3);
add(b4);
add(t1);
add(t2);
add(t3);
add(t4);
}
public static void main(String[] args) {
Gridex fl = new Gridex("Grid Layout demo");
fl.setSize(400,400);
fl.setVisible(true);
}
```

}

}

Output:-

```
C:\Users\ADMIN\Desktop>cd siva_java_files

C:\Users\ADMIN\Desktop\siva_java_files>javac Gridex

C:\Users\ADMIN\Desktop\siva_java_files>java Gridex
```



Program:- 11

```
import java.awt.*;
import java.awt.event.*;
class BEventEx extends Frame implements ActionListener
{
Button bshow,bclose;
BEventEx(String tit)
{
super(tit);
bshow = new Button ("show");
bclose = new Button ("close");
bshow.addActionListener(this);
bclose.addActionListener(this);
setLayout(new FlowLayout());
add(bshow);
add(bclose);
}
public void actionPerformed(ActionEvent ae)
{
if (ae.getSource().equals(bshow))
System.out.println("Show Button is clicked");
else if(ae.getSource().equals(bclose))
System.exit(0);
}

public static void main(String[] args)
{
BEventEx b= new BEventEx("Event Demo");
b.setSize(300,400);
b.setVisible(true);
}
```

Output:-

```
C:\Users\ADMIN\Desktop>cd siva_java_files

C:\Users\ADMIN\Desktop\siva_java_files>javac BEventEx.java

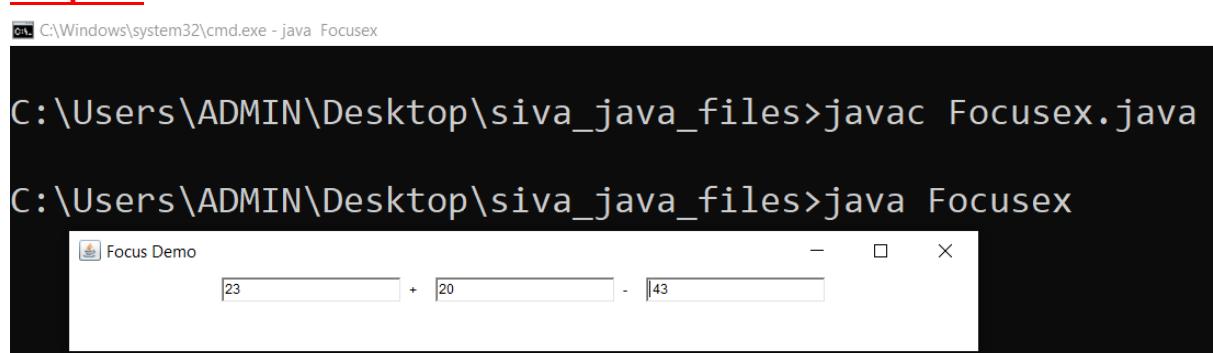
C:\Users\ADMIN\Desktop\siva_java_files>java BEventEx
Show Button is clicked
```



Program:- 12

```
import java.awt.*;
import java.awt.event.*;
class Focusex extends Frame implements FocusListener
{
TextField t1,t2,t3;
Label label1,label2;
Focusex(String tit)
{
super(tit);
t1 = new TextField(20);
t2 = new TextField(20);
t3 = new TextField(20);
label1 = new Label("+");
label2 = new Label("-");
t3.addFocusListener(this);
setLayout(new FlowLayout());
add(t1);
add(label1);
add(t2);
add(label2);
add(t3);
}
public void focusGained(FocusEvent fe)
{
int a = Integer.parseInt(t1.getText());
int b = Integer.parseInt(t2.getText());
t3.setText(" "+(a+b));
}
public void focusLost(FocusEvent fe)
{
}
}
public static void main(String [] args)
{
Focusex fe = new Focusex("Focus Demo");
fe.setSize(500,600);
fe.setVisible(true);
}
```

Output:-



C:\Windows\system32\cmd.exe - java Focusex

C:\Users\ADMIN\Desktop\siva_java_files>javac Focusex.java

C:\Users\ADMIN\Desktop\siva_java_files>java Focusex

Focus Demo

23 + 20 - 43

Program:- 13

Java program to create Menu Bar

```
import java.awt.*;
import java.awt.event.*;
class Menuex extends Frame implements ActionListener
{
MenuBar mb;
Menu mfile,medit;
MenuItem minew,miopen,miexit;
Menuex(String title)
{
super(title);
mb = new MenuBar();
mfile = new Menu("File");
medit = new Menu("Edit");
minew = new MenuItem("New");
miopen = new MenuItem("Open");
miexit = new MenuItem("Exit");
minew.addActionListener(this);
miopen.addActionListener(this);
miexit.addActionListener(this);
mfile.add(minew);
mfile.add(miopen);
mfile.addSeparator();
mfile.add(miexit);
mb.add(mfile);
mb.add(medit);
setMenuBar(mb);
}
public void actionPerformed(ActionEvent ae)
{
if (ae.getSource().equals(minew))
{
System.out.println("Selected new");
}
if(ae.getSource().equals(miopen))
{
System.out.println("Selected open");
}
```

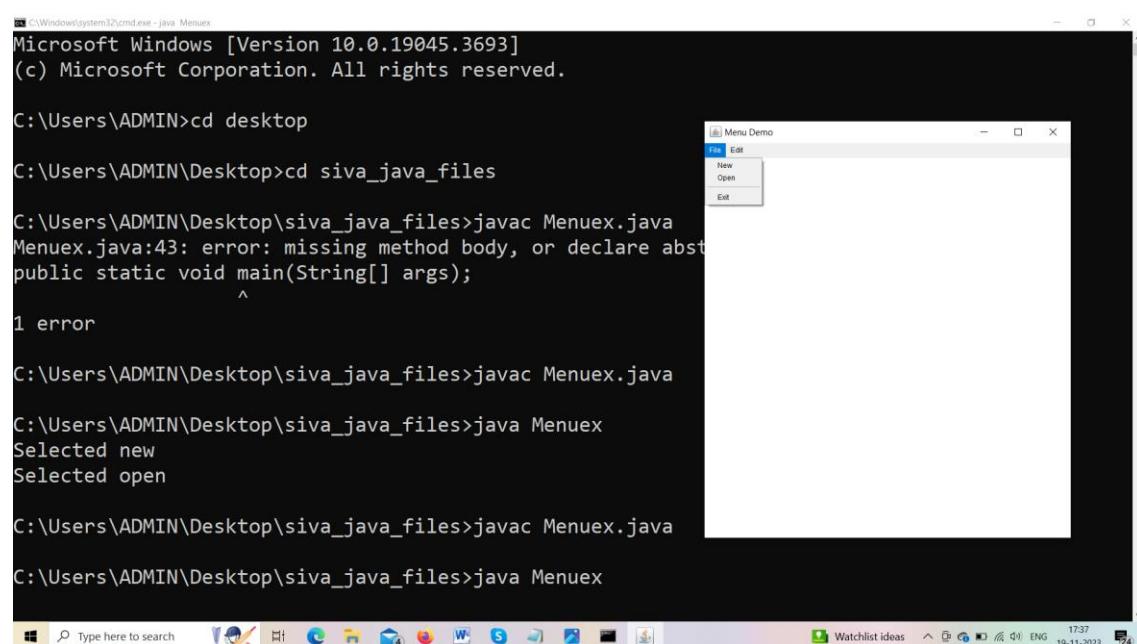
```

if (ae.getSource().equals(miexit))
{
    System.exit(0);
}
}

public static void main(String[] args)
{
    Menuex fe = new Menuex("Menu Demo");
    fe.setSize(500,600);
    fe.setVisible(true);
}
}

```

Output:-



The image shows a Windows command prompt window and a Java application window side-by-side.

Command Prompt Output:

```

C:\Windows\system32\cmd.exe - java.Menuex
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>cd desktop
C:\Users\ADMIN\Desktop>cd siva_java_files
C:\Users\ADMIN\Desktop\siva_java_files>javac Menuex.java
Menuex.java:43: error: missing method body, or declare abstract
public static void main(String[] args);
^
1 error

C:\Users\ADMIN\Desktop\siva_java_files>javac Menuex.java

C:\Users\ADMIN\Desktop\siva_java_files>java Menuex
Selected new
Selected open

C:\Users\ADMIN\Desktop\siva_java_files>javac Menuex.java

C:\Users\ADMIN\Desktop\siva_java_files>java Menuex

```

Java Application Screenshot:

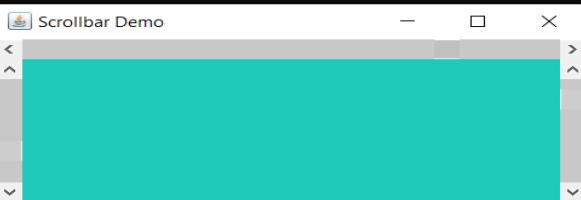
A window titled "Menu Demo" is shown. A context menu is open over the window, displaying options: File, Edit, New, Open, and Exit. The "File" option is highlighted.

Program:- 14

Java program to create a Scroll bar

```
import java.awt.*;
import java.awt.event.*;
class Scrollex extends Frame implements AdjustmentListener
{
Scrollbar s1,s2,s3;
Color cl;
Scrollex(String tit)  {
super(tit);
s1 = new Scrollbar(Scrollbar.VERTICAL,0,5,0,255);
s2 = new Scrollbar(Scrollbar.HORIZONTAL,0,5,0,255);
s3 = new Scrollbar(Scrollbar.VERTICAL,0,5,0,255);
s1.addAdjustmentListener(this);
s2.addAdjustmentListener(this);
s3.addAdjustmentListener(this);
add(s1, BorderLayout.EAST);
add(s2, BorderLayout.NORTH);
add(s3, BorderLayout.WEST);
}
public void adjustmentValueChanged(AdjustmentEvent ae)
{
repaint();
}
public void paint(Graphics g)  {
cl = new Color(s1.getValue(),s2.getValue(),s3.getValue());
setBackground(cl);
}
public static void main(String[] args)  {
Scrollex fe = new Scrollex("Scrollbar Demo");
fe.setSize(500,600);
fe.setVisible(true);
}
```

Output:-

```
C:\Windows\system32\cmd.exe - java Scrollex
C :\Users\ADMIN\Desktop\siva_java_files>javac Scrollex.java
C :\Users\ADMIN\Desktop\siva_java_files>java Scrollex

```

Program:- 15

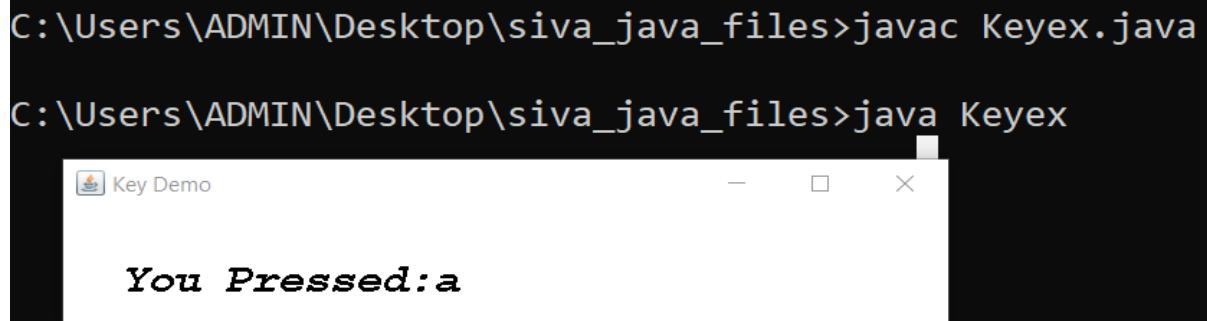
Java program to create a Key Event

```
import java.awt.*;
import java.awt.event.*;
class Keyex extends Frame implements KeyListener {
    String k1;
    Keyex(String tit) {
        super(tit);
        k1 = "";
        addKeyListener(this);
    }
    public void keyPressed(KeyEvent ke) {
        k1 = "You Pressed:" + ke.getKeyCode();
        repaint();
    }
    public void keyTyped(KeyEvent ke) {
        k1 = "You Pressed:" + ke.getKeyChar();
        repaint();
    }
    public void keyReleased(KeyEvent ae) {
        repaint();
    }
    public void paint(Graphics g) {
        Font f = new Font("Courier", Font.BOLD + Font.ITALIC, 30);
        g.setFont(f);
        g.drawString(k1, 50, 100);
    }
    public static void main(String[] args) {
        Keyex fe = new Keyex("Key Demo");
        fe.setSize(500, 600);
        fe.setVisible(true);
    }
}
```

Output:-

```
C:\Users\ADMIN\Desktop\siva_java_files>javac Keyex.java

C:\Users\ADMIN\Desktop\siva_java_files>java Keyex
```



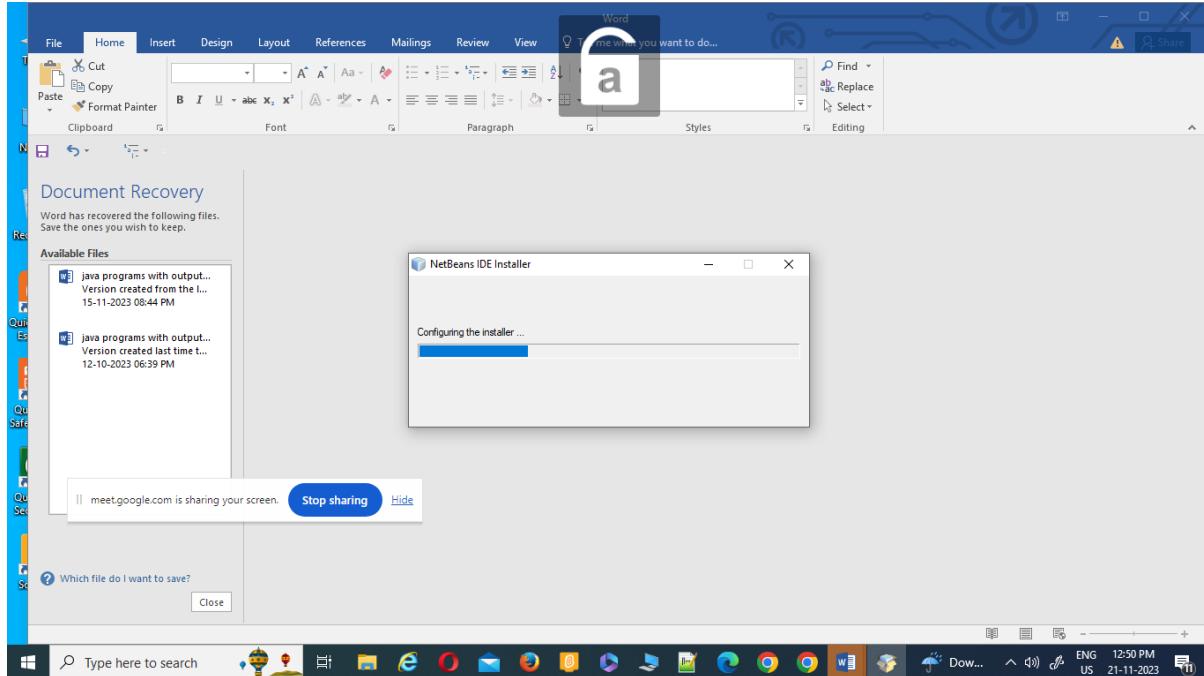
The screenshot shows a Java application window titled "Key Demo". The window has a black background and contains the text "You Pressed:a" in white. The window has standard operating system controls (minimize, maximize, close) at the top right.

MINI PROJECT

SOFTWARE INSTALLATION:-

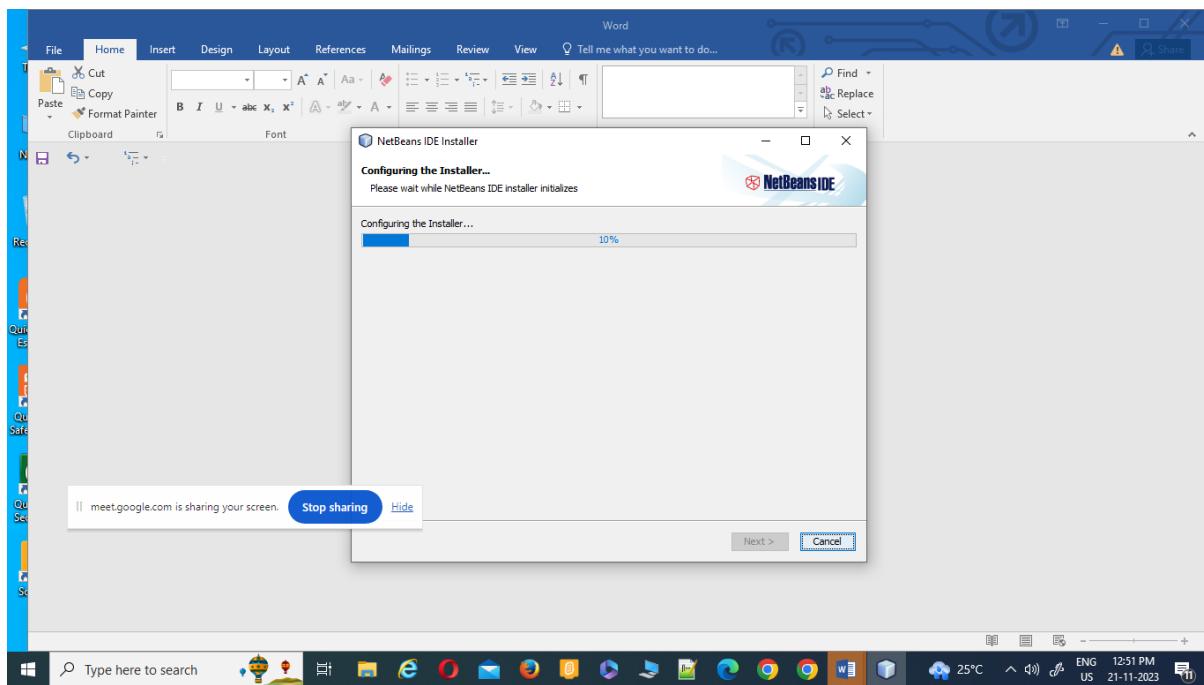
Double click on netbeans downloaded software

We will get the below shown picture

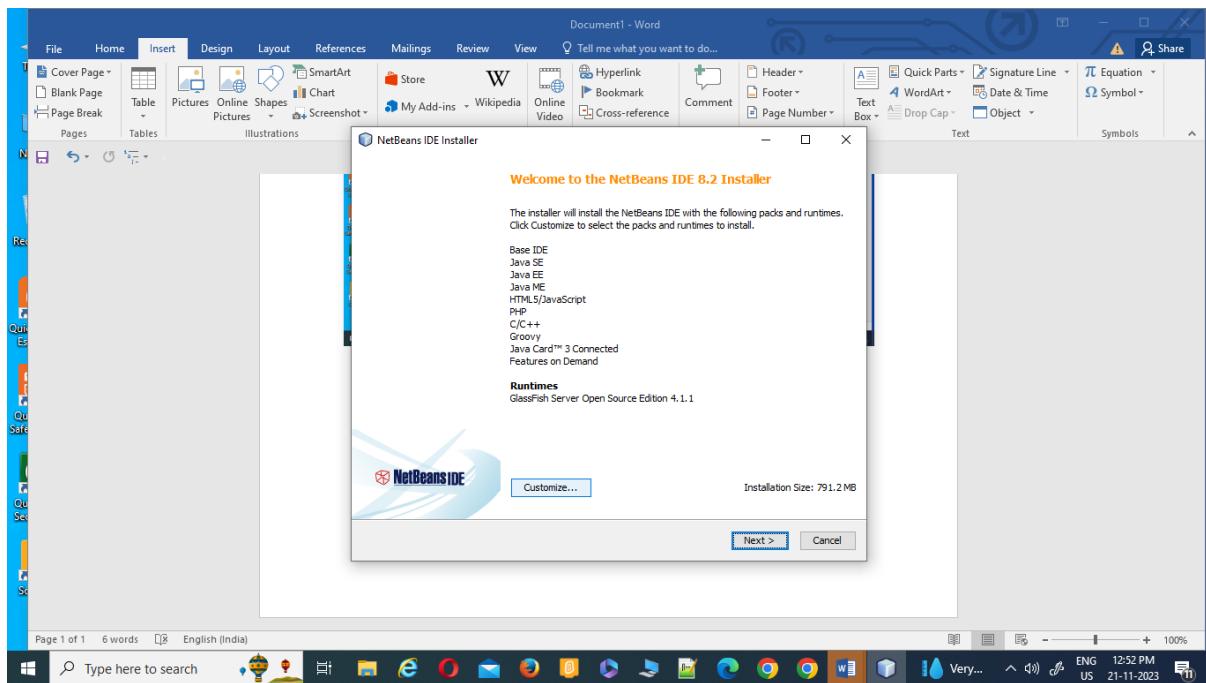


Make sure it should be completed 100%

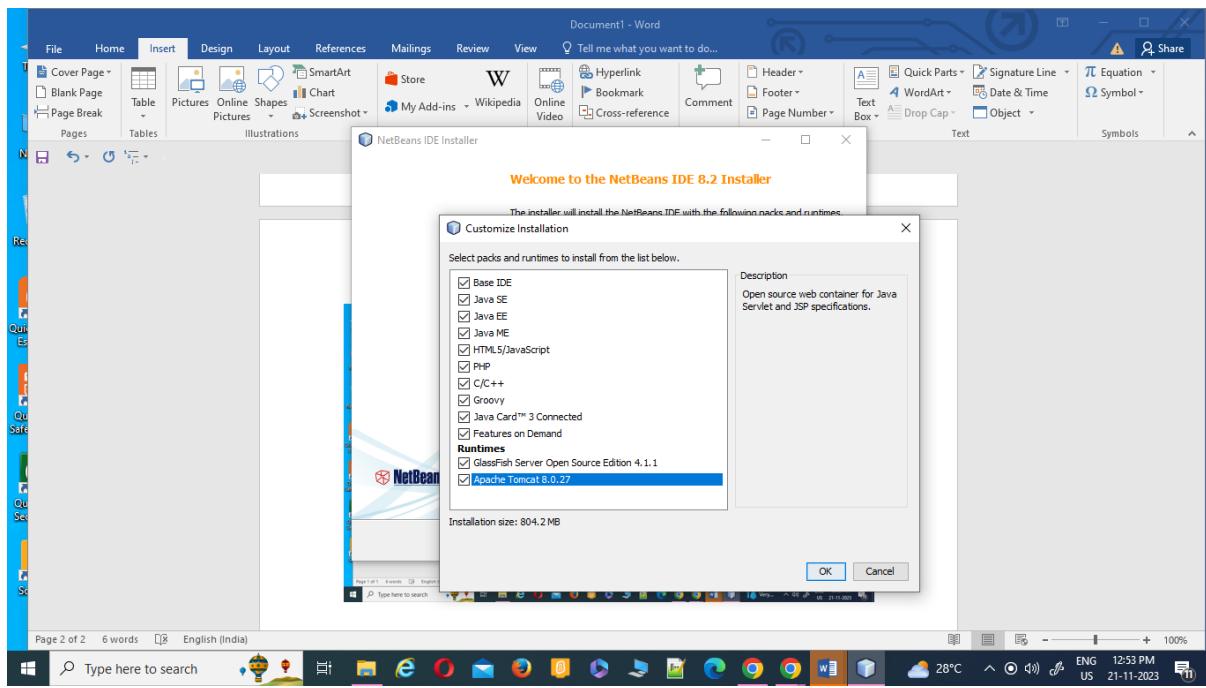
Then we will get the below shown screenshot

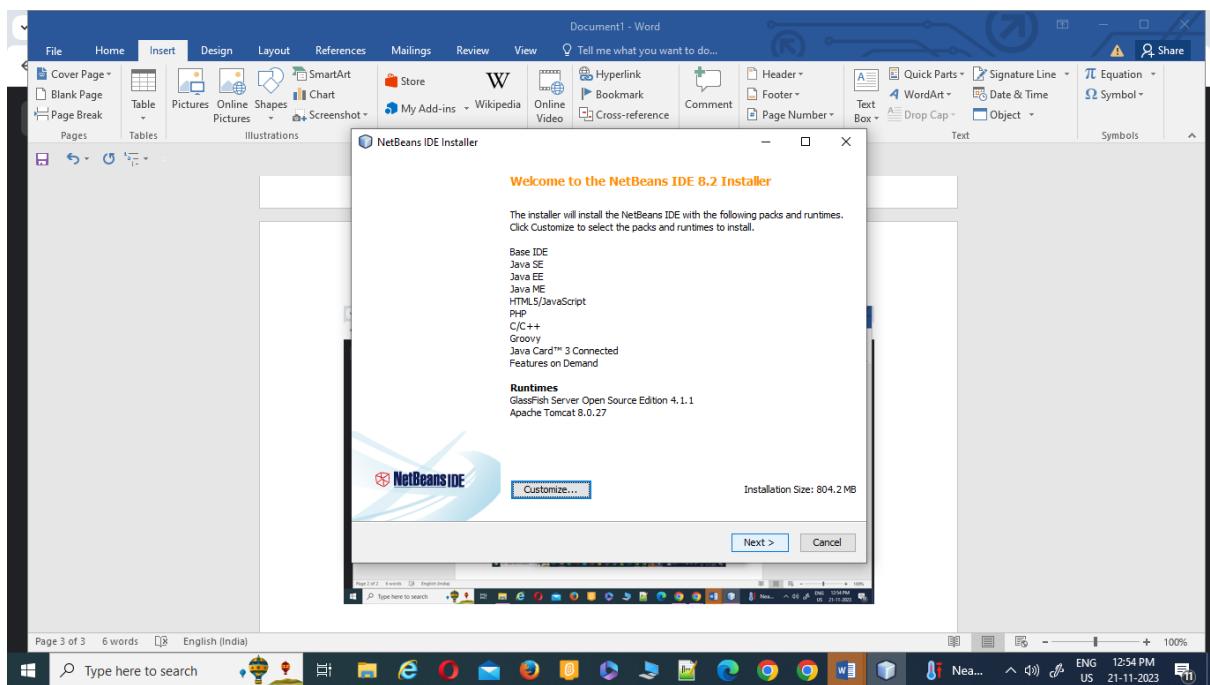
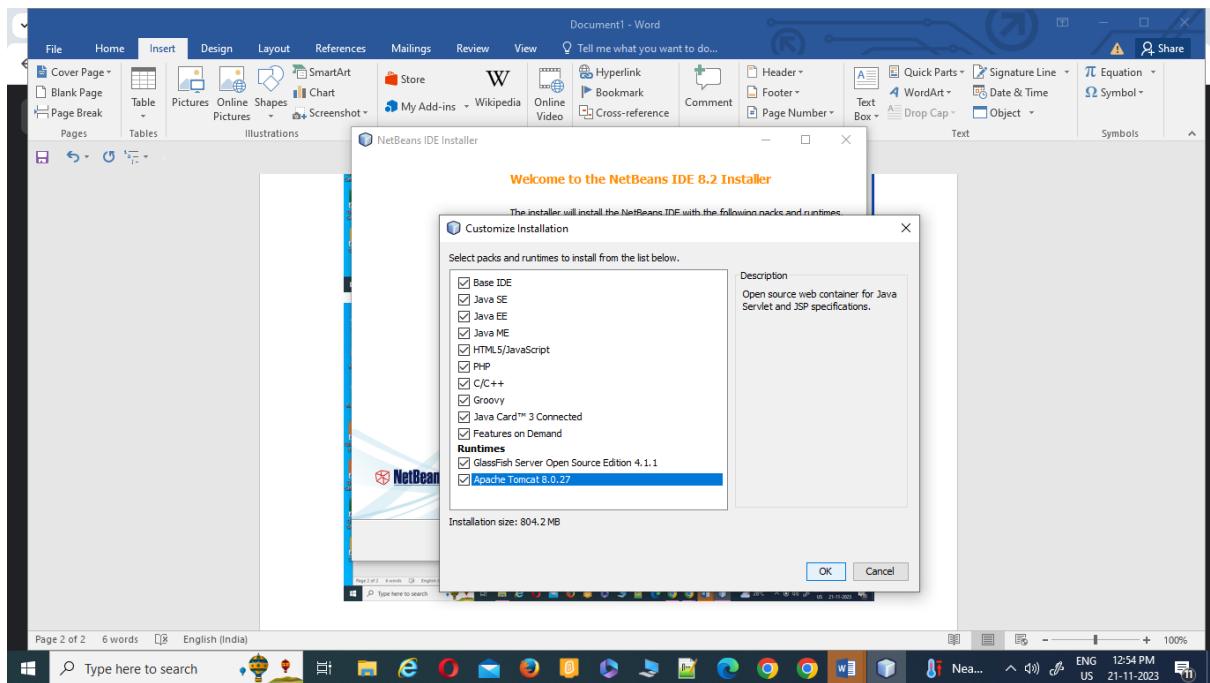


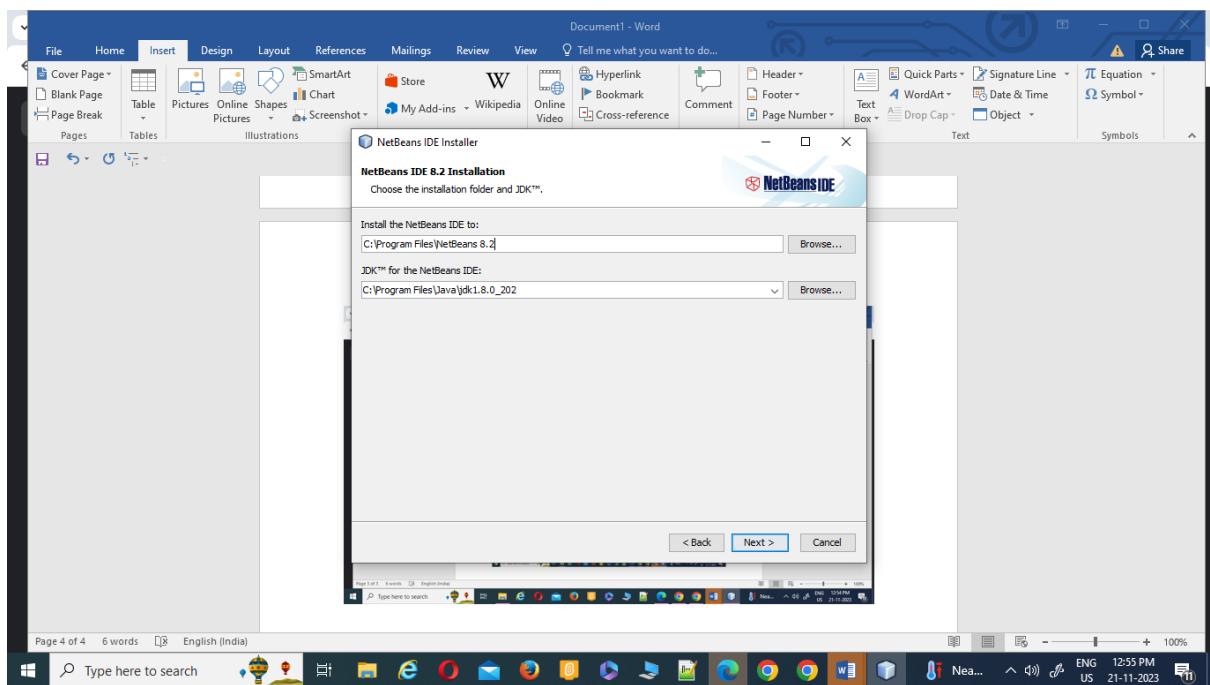
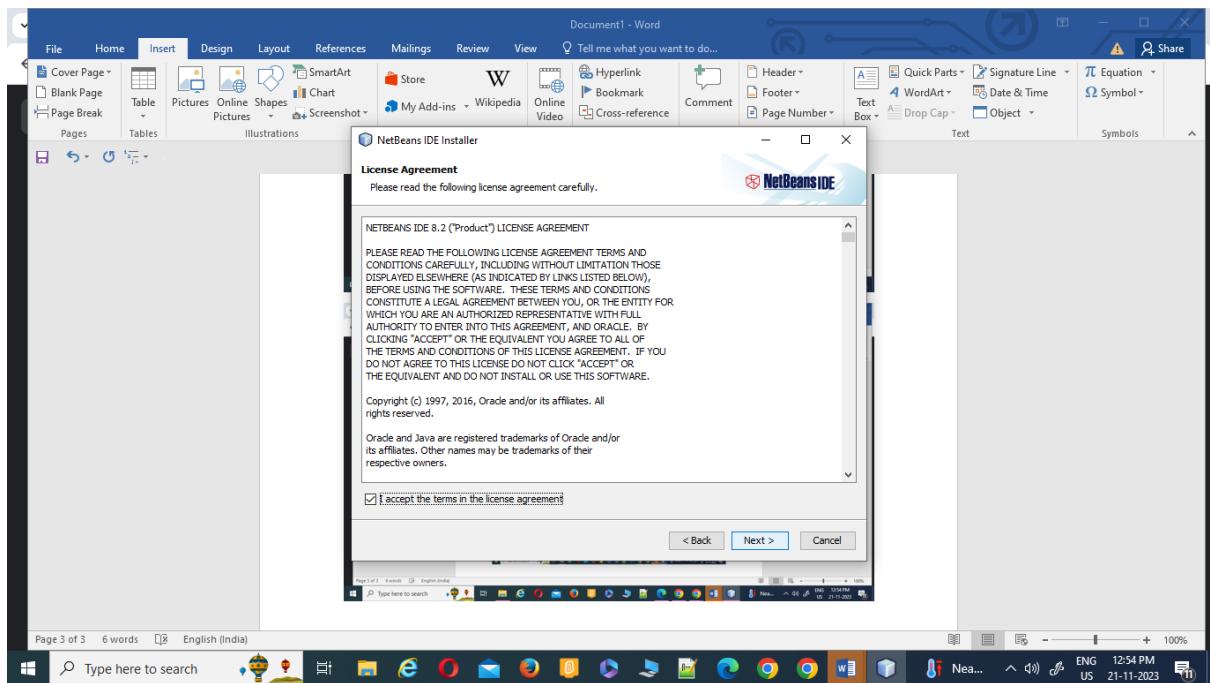
Then click on customize button.

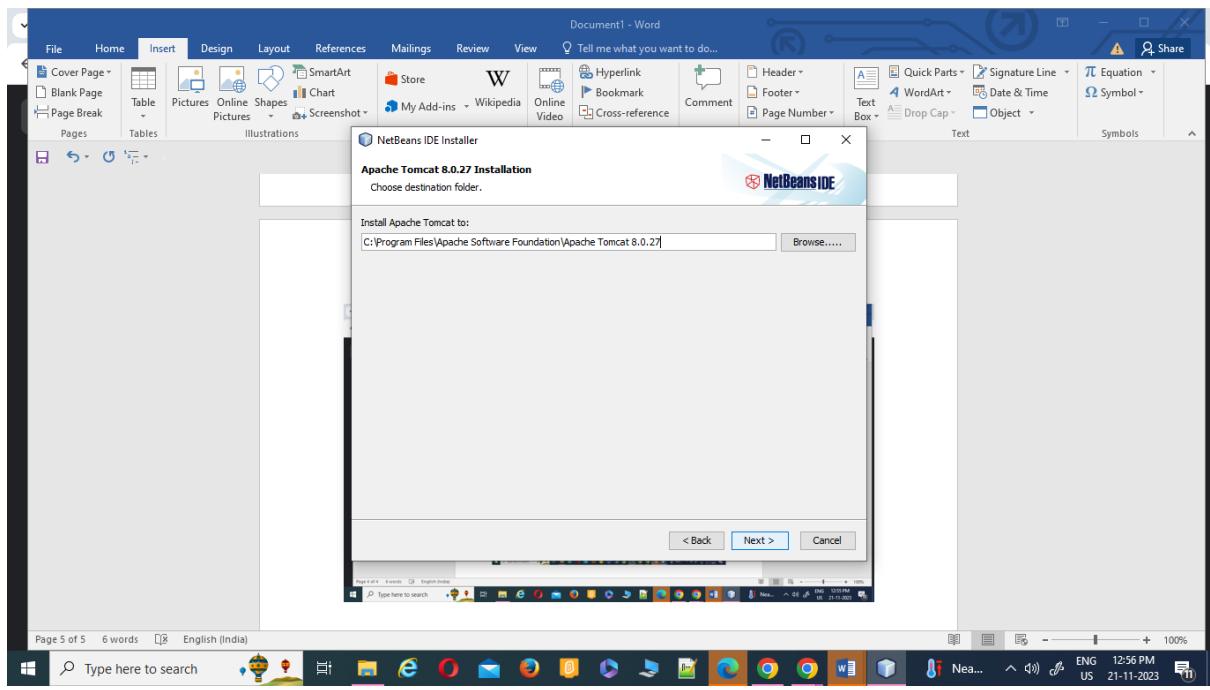
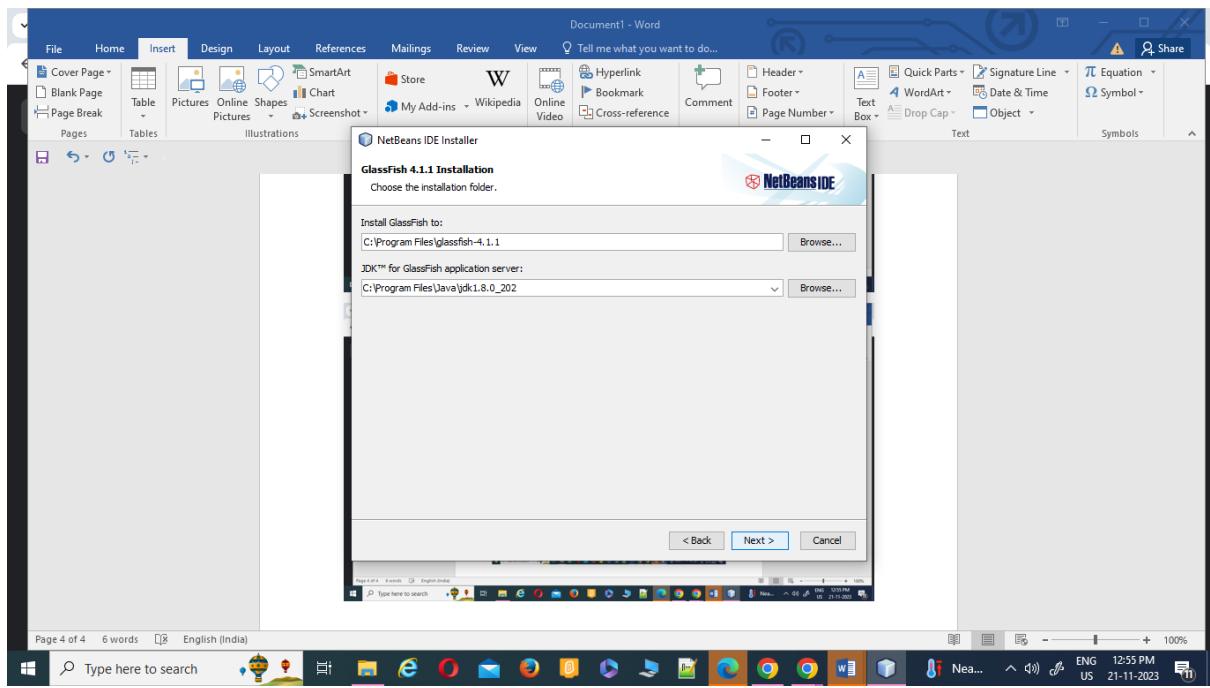


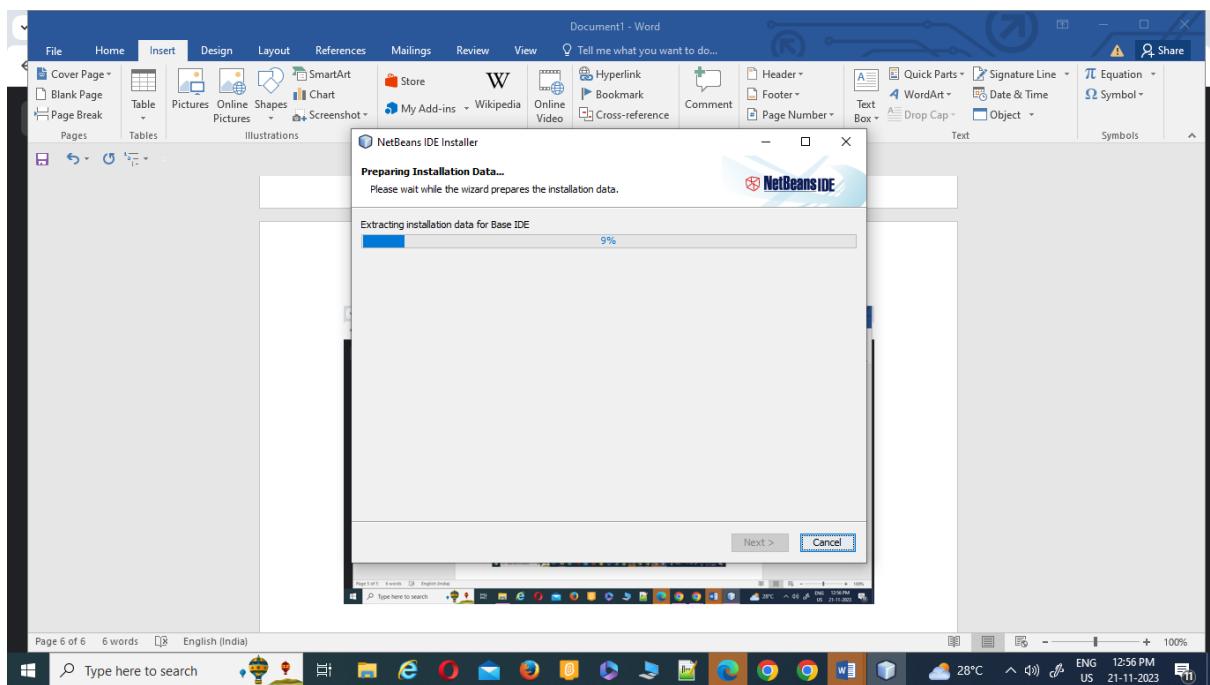
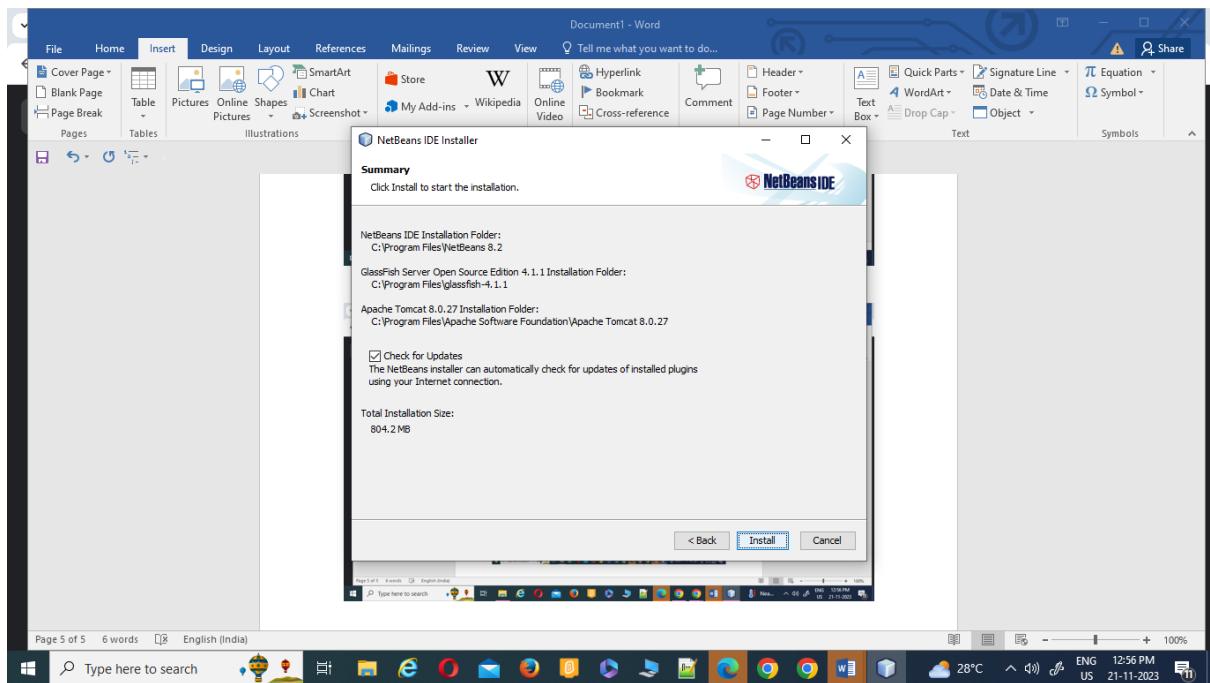
Then select checkbox Apache Tomcat Server and then click on ok button.

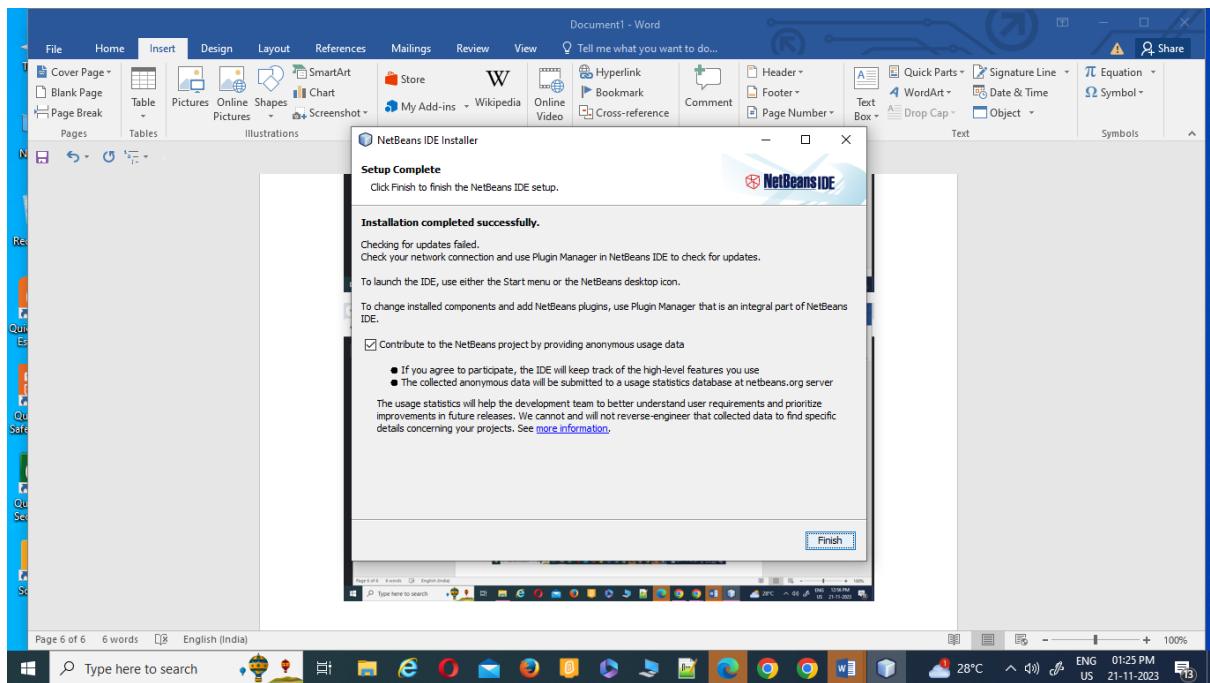






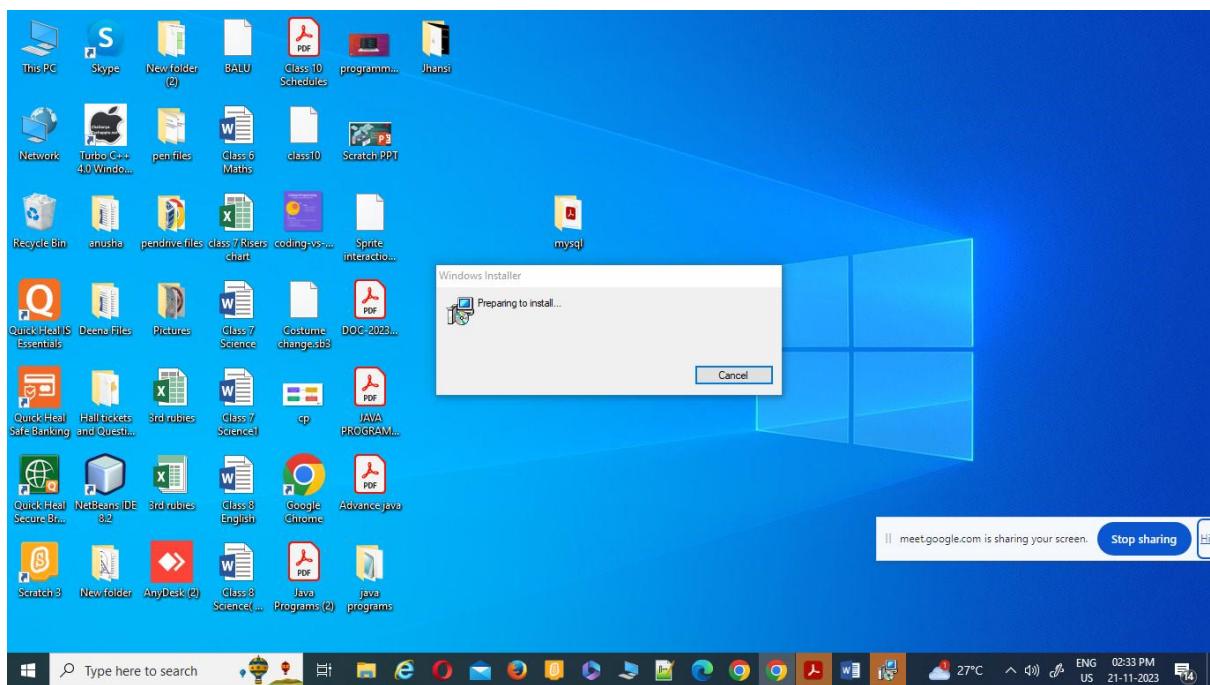


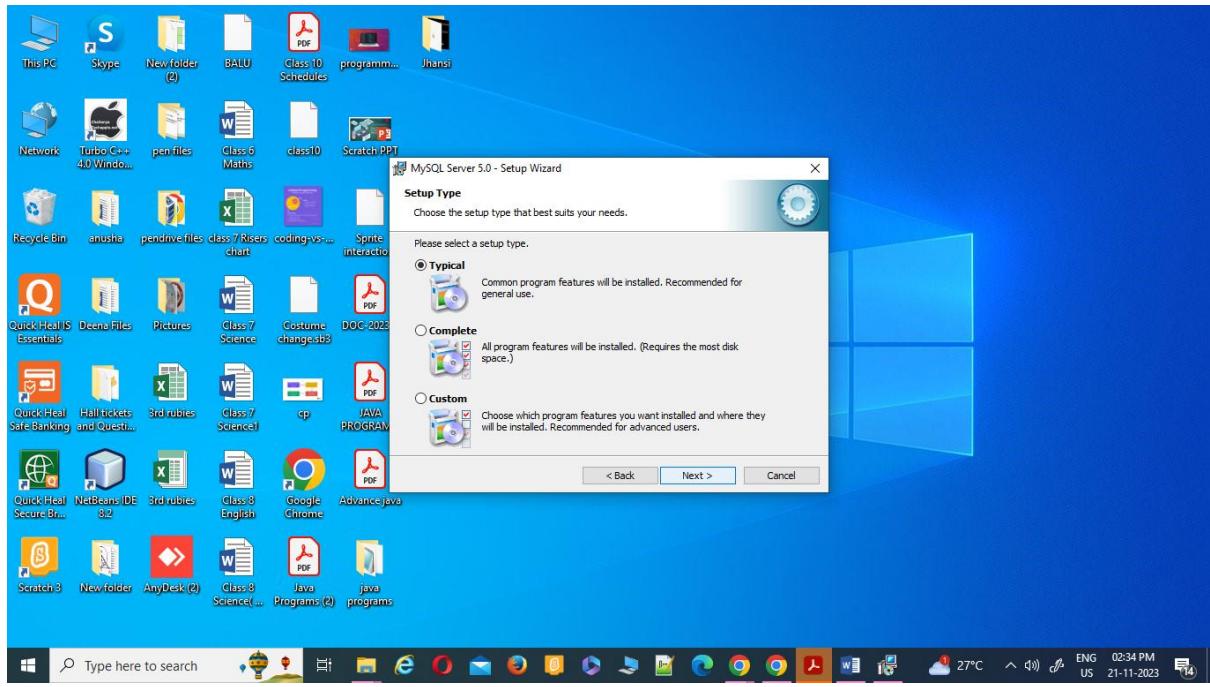
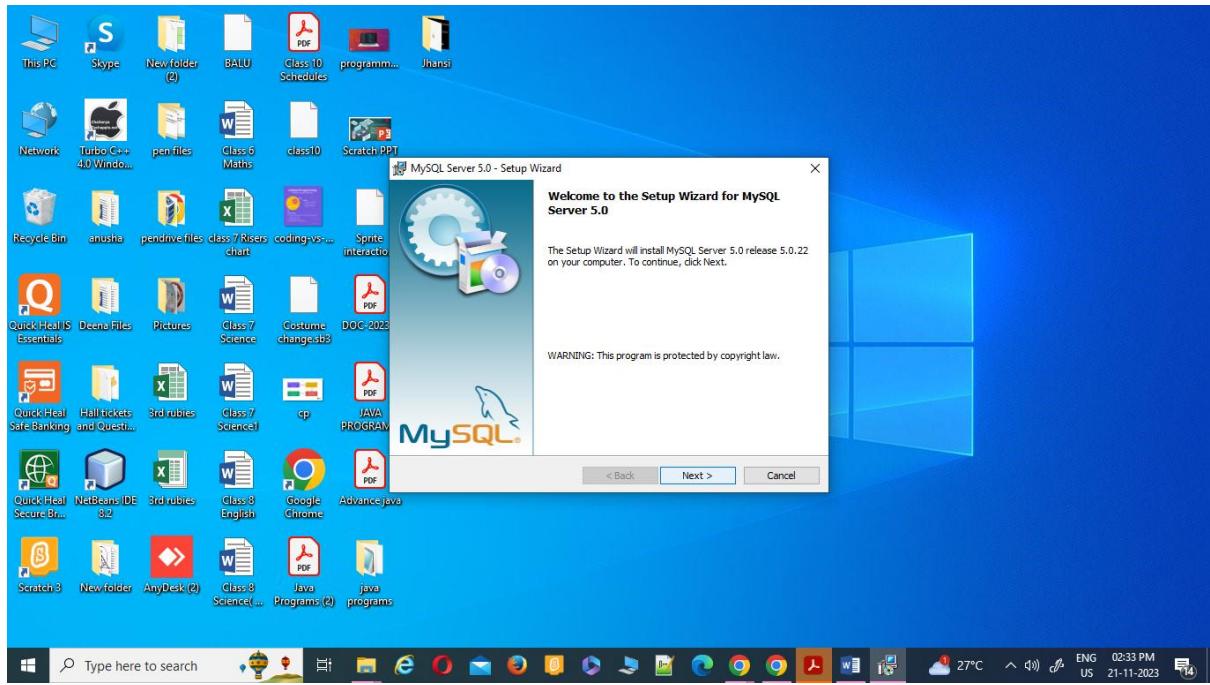


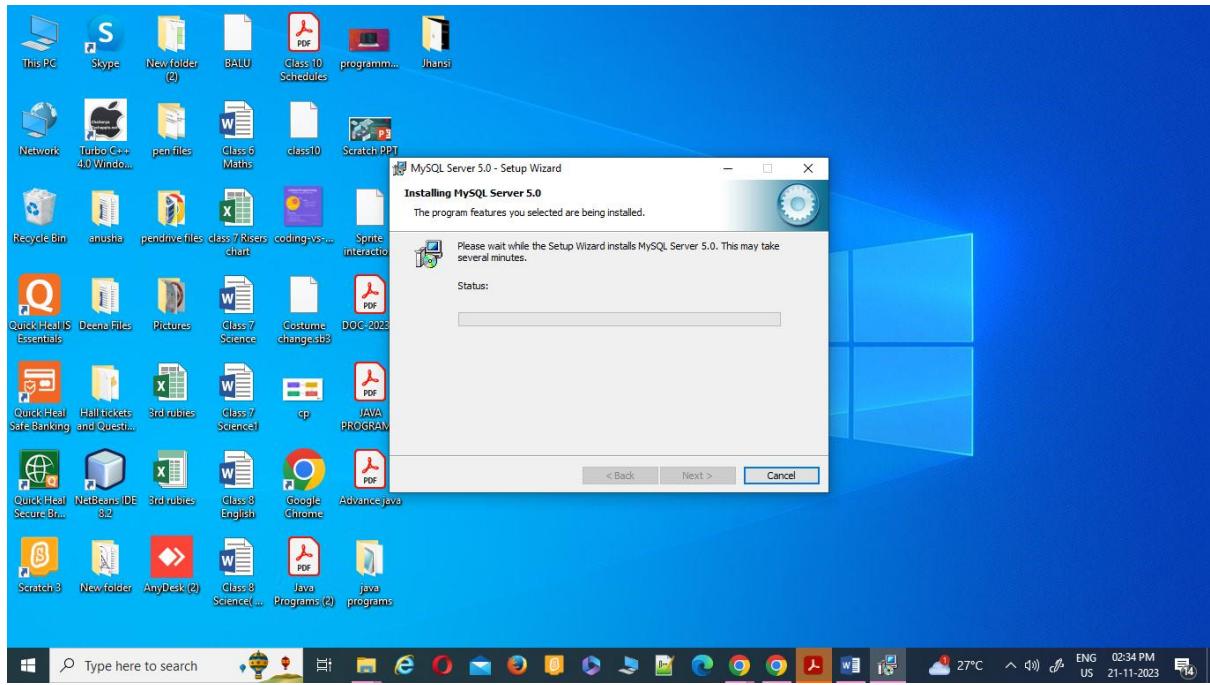
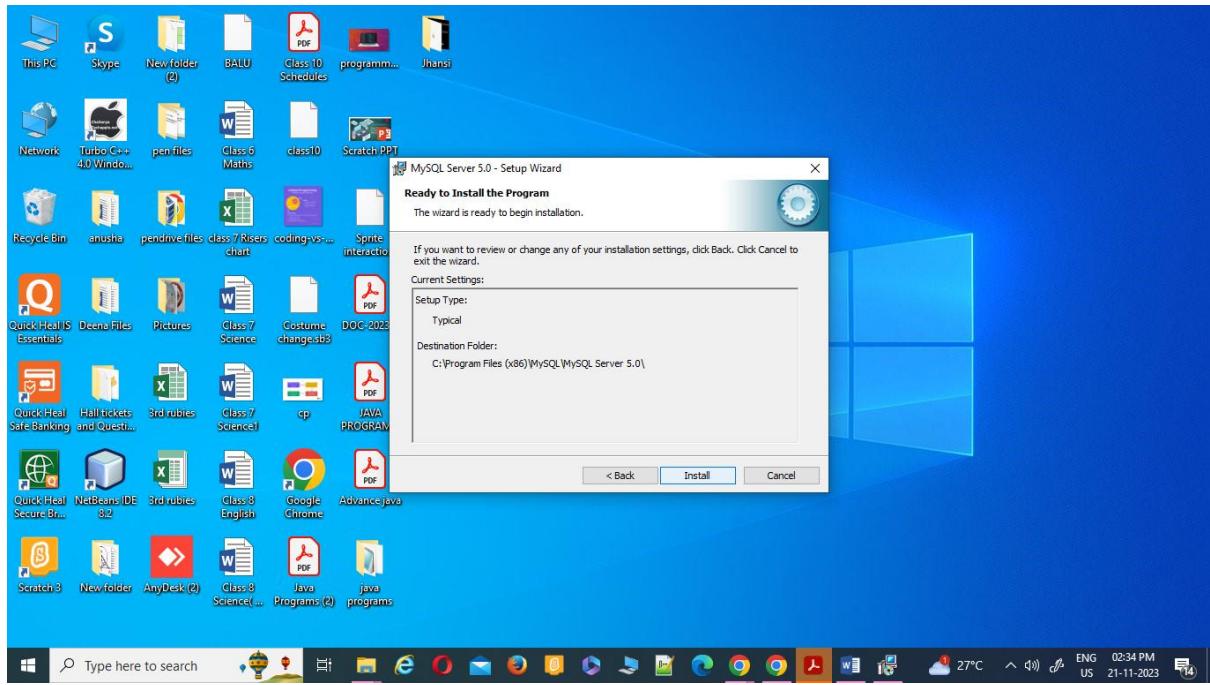


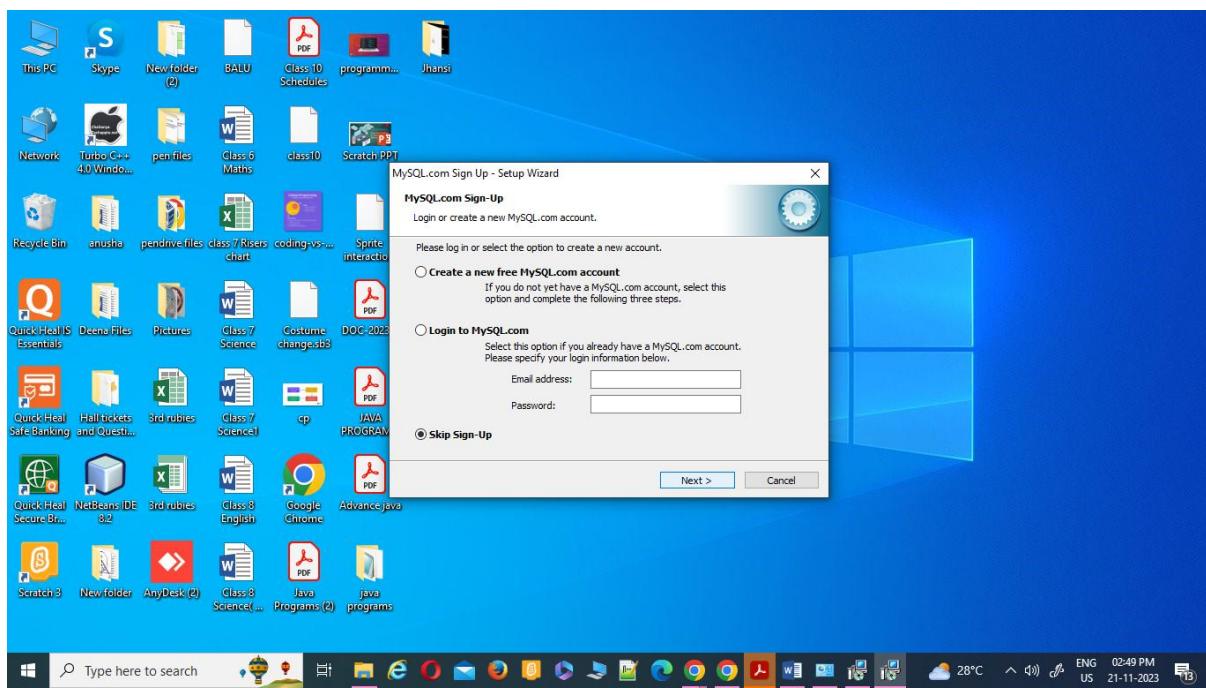
Procedure to install my sql

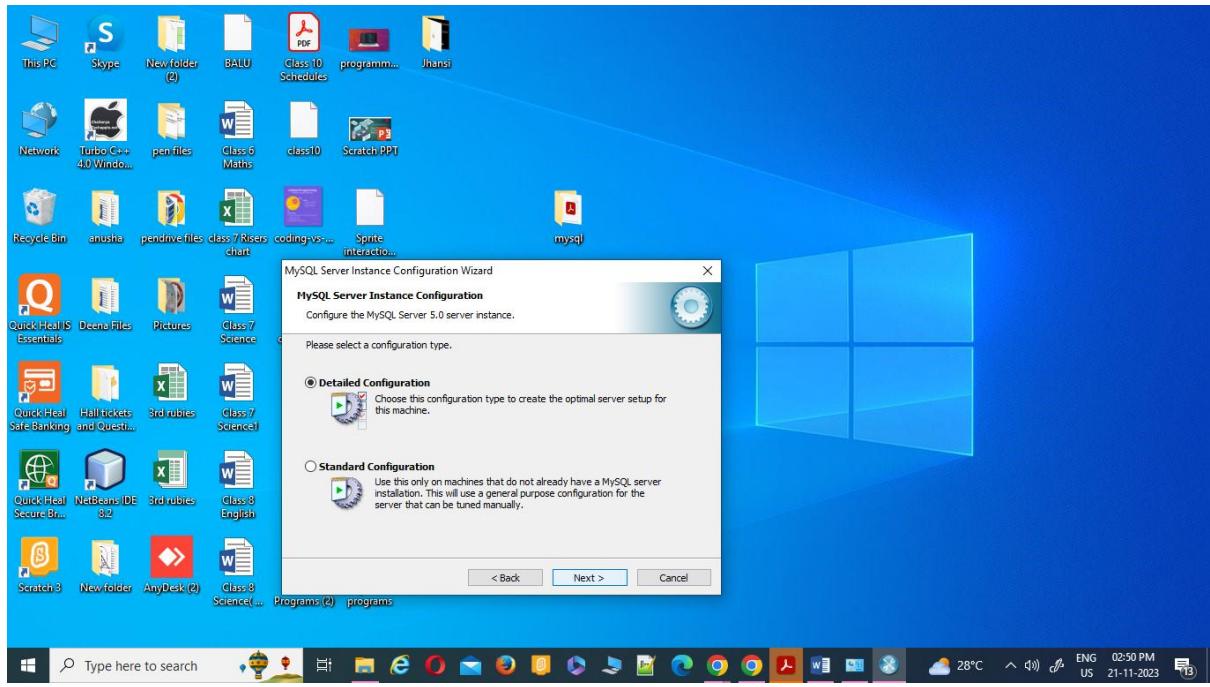
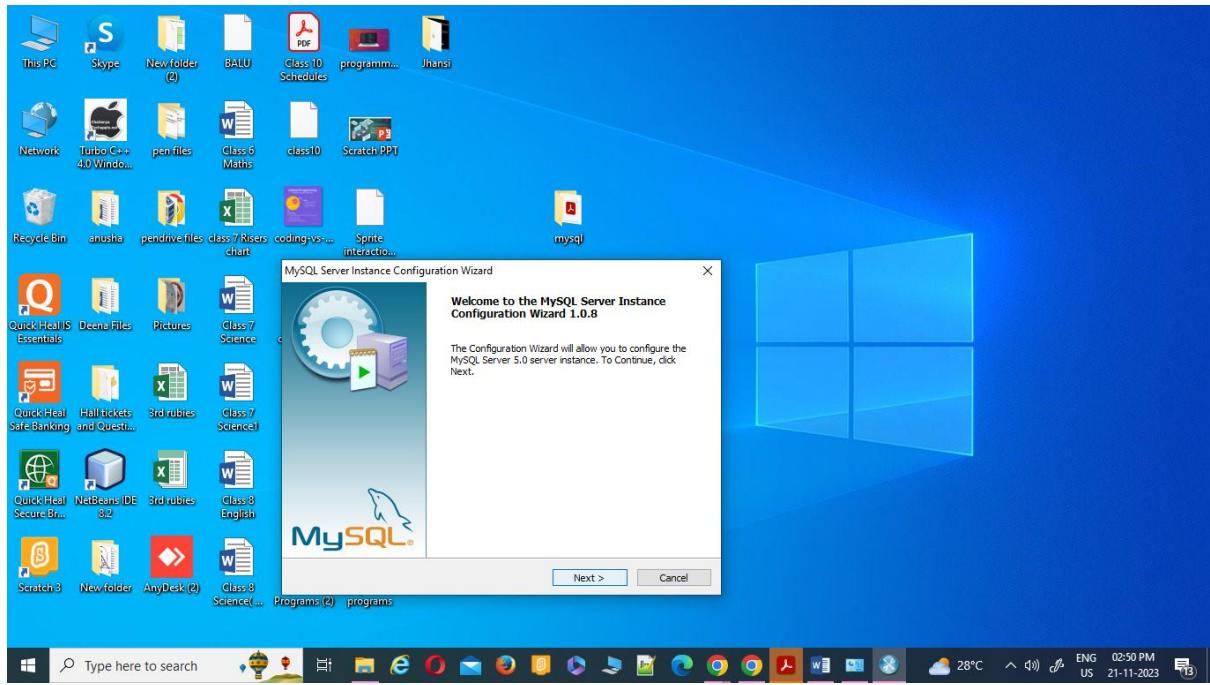
Setup file

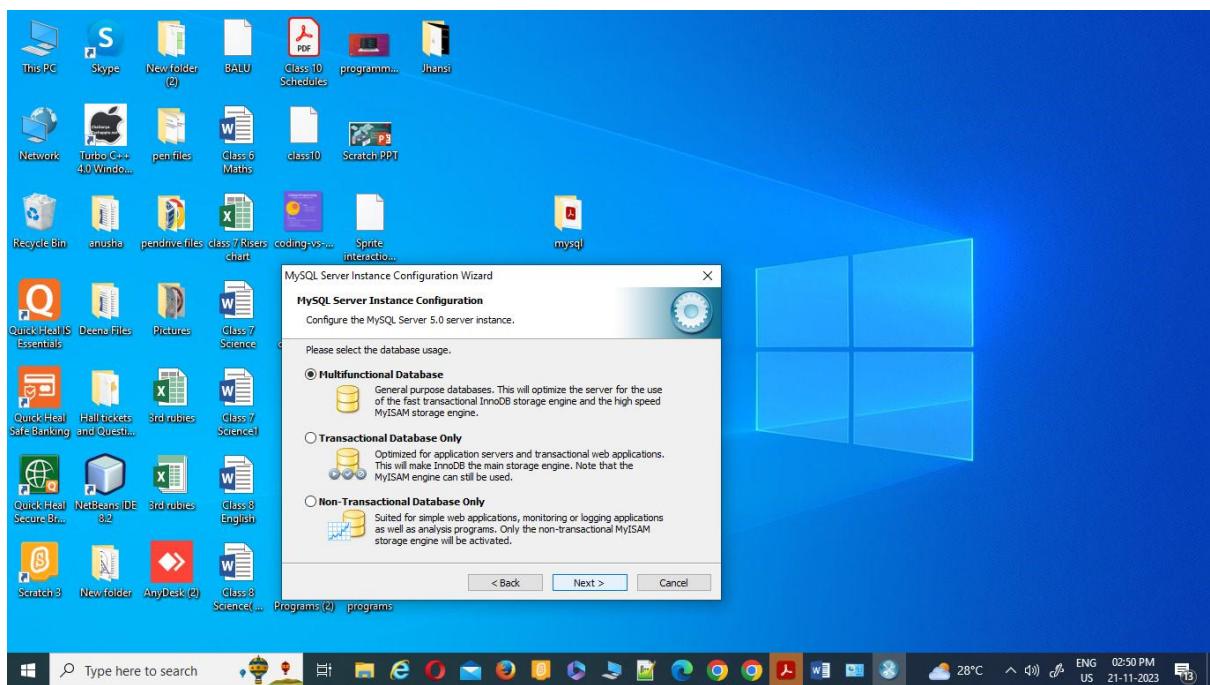
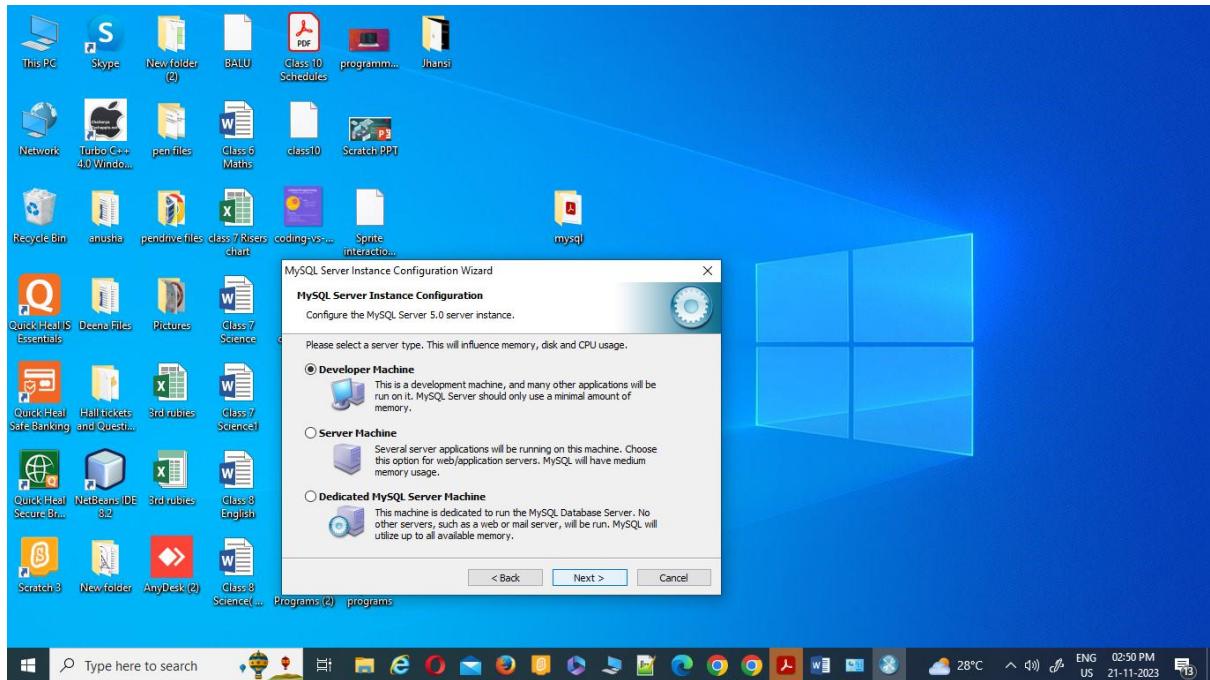


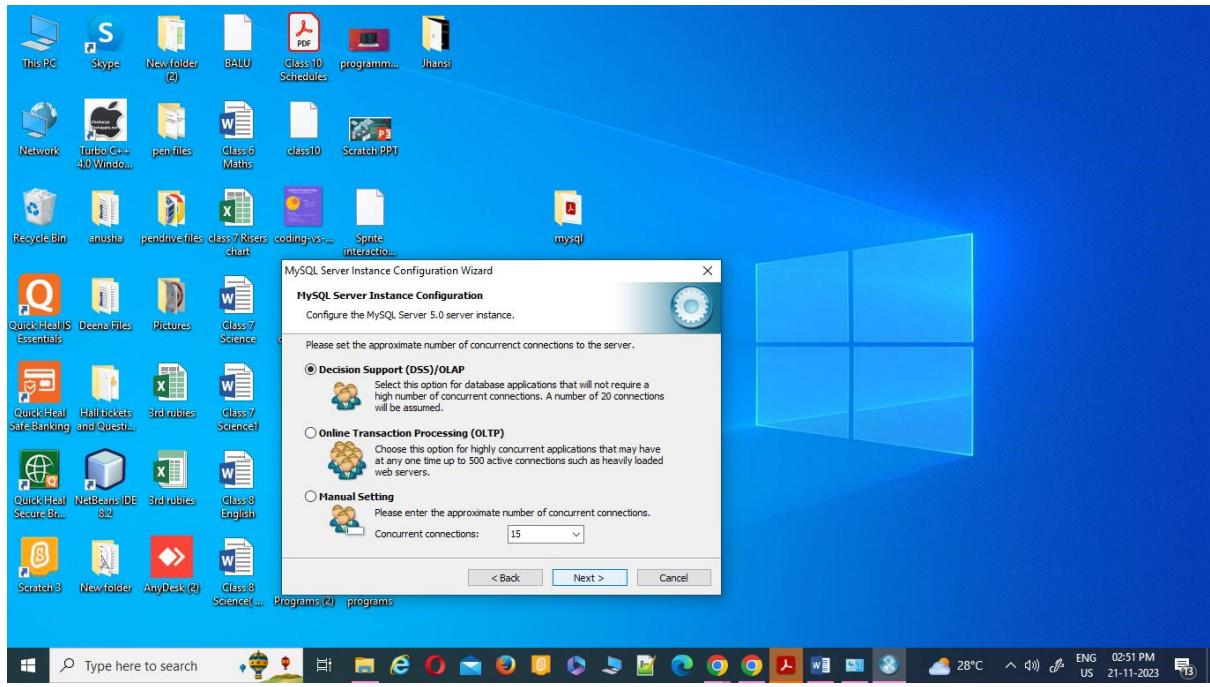
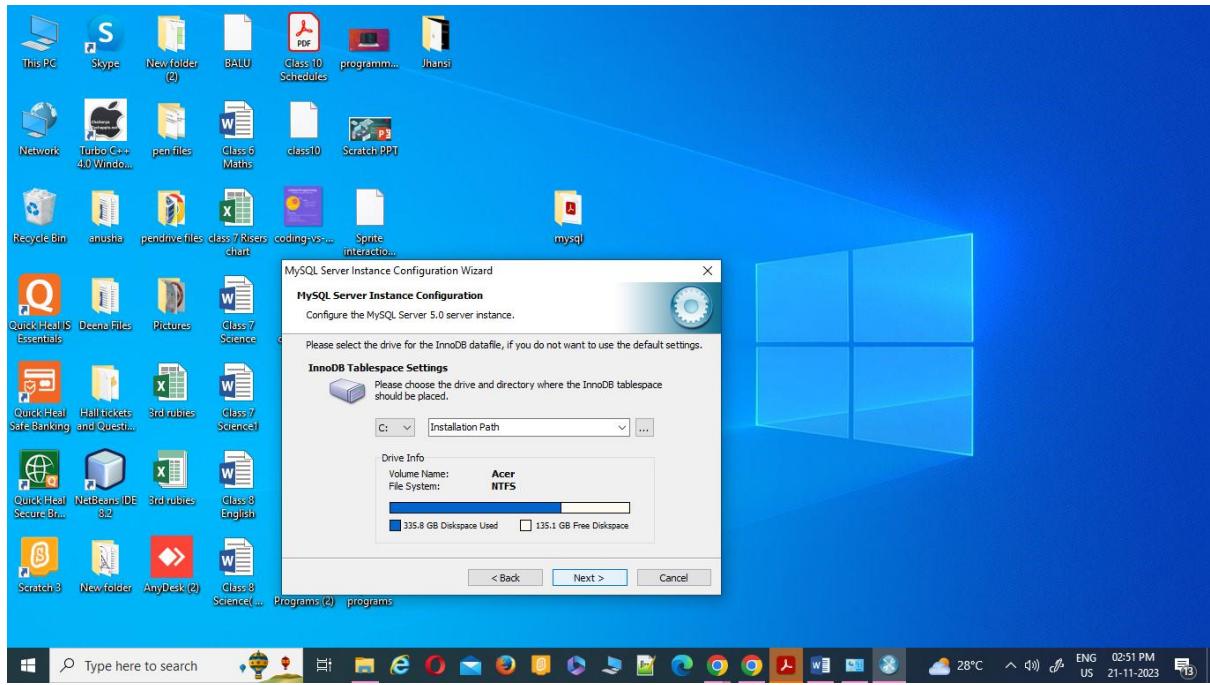


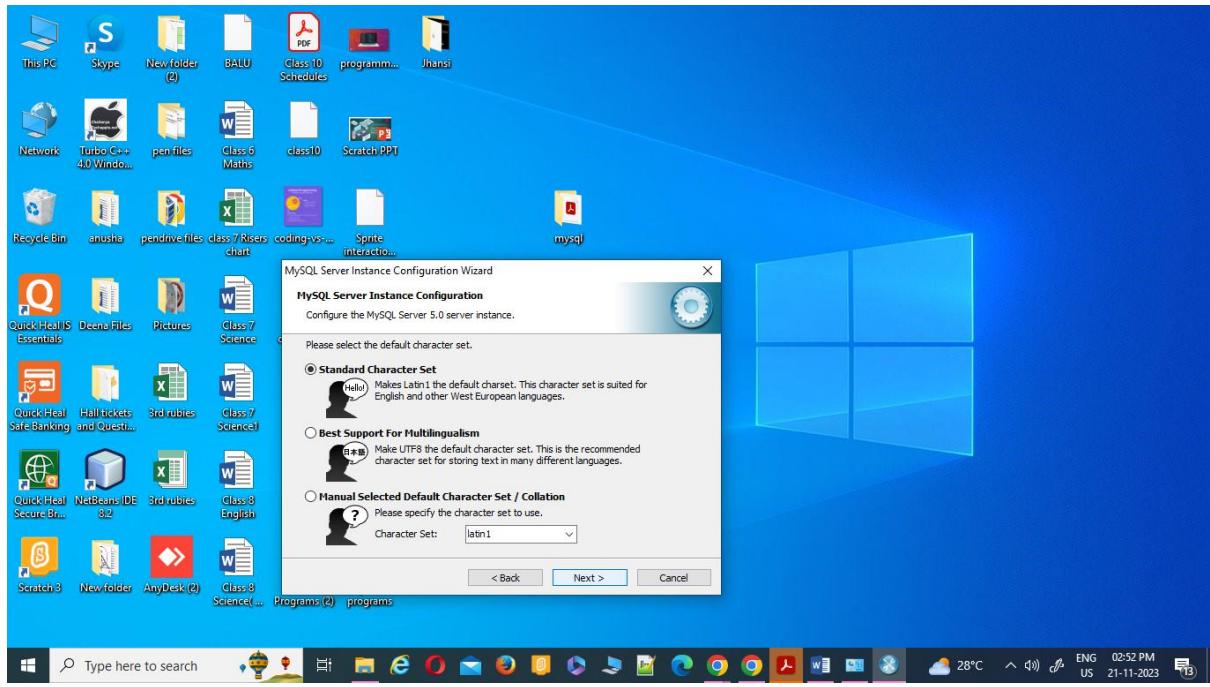
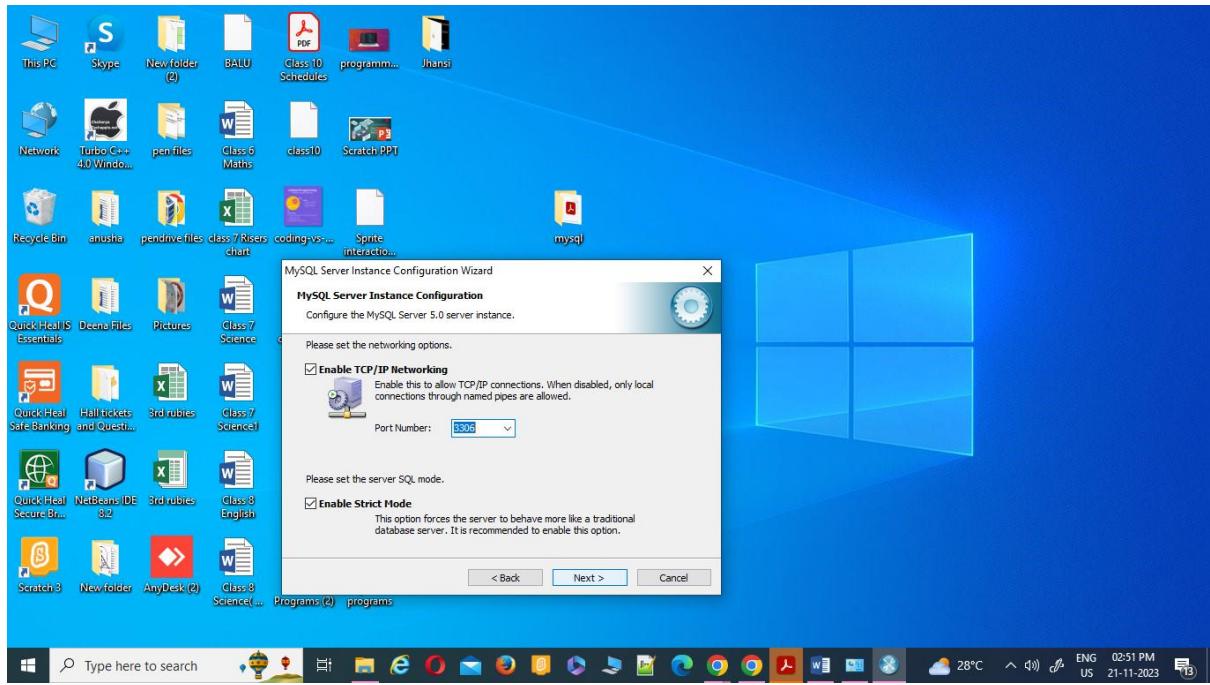


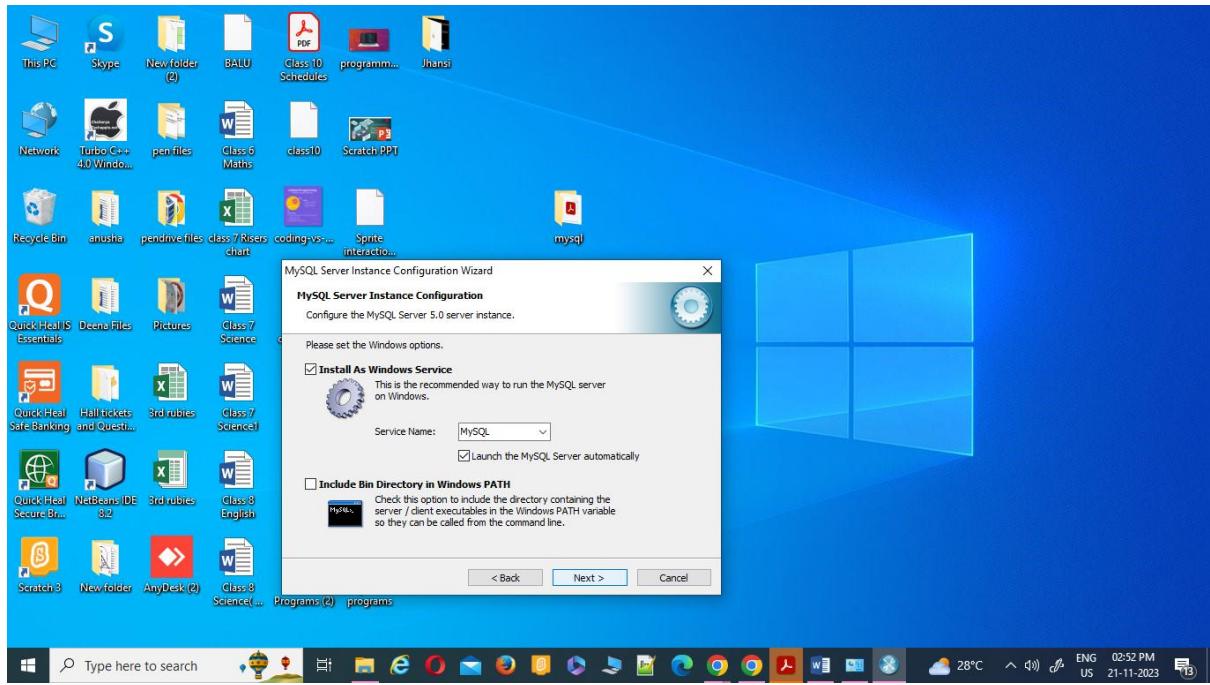




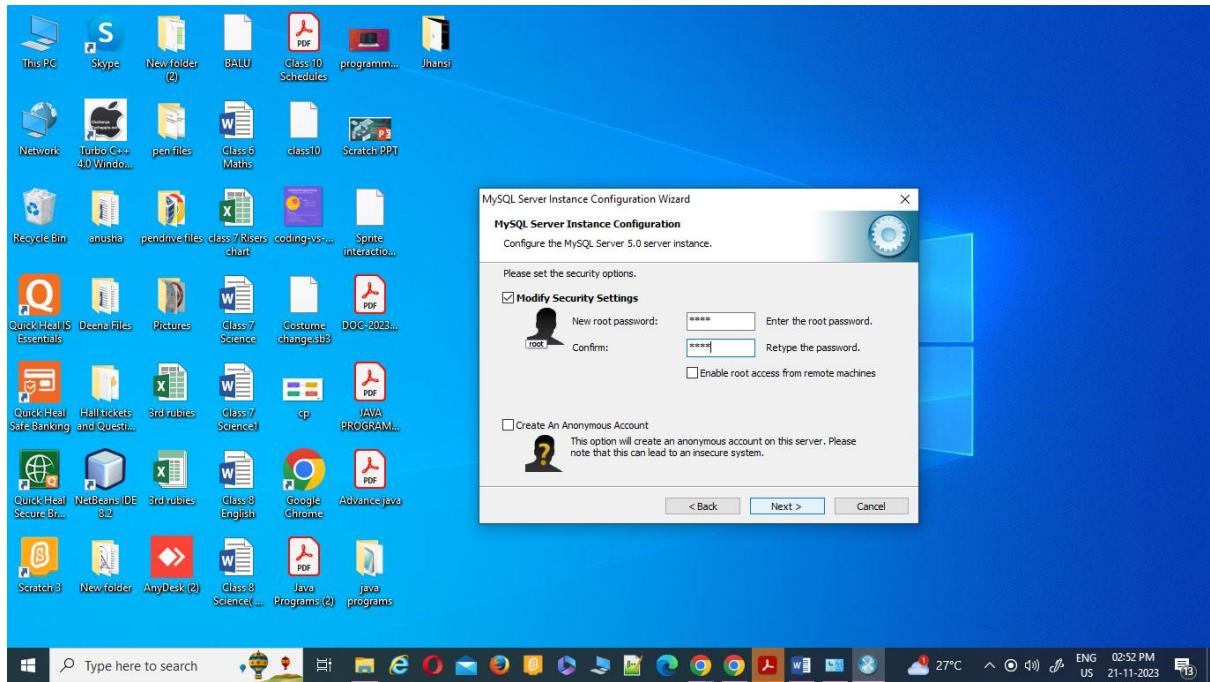


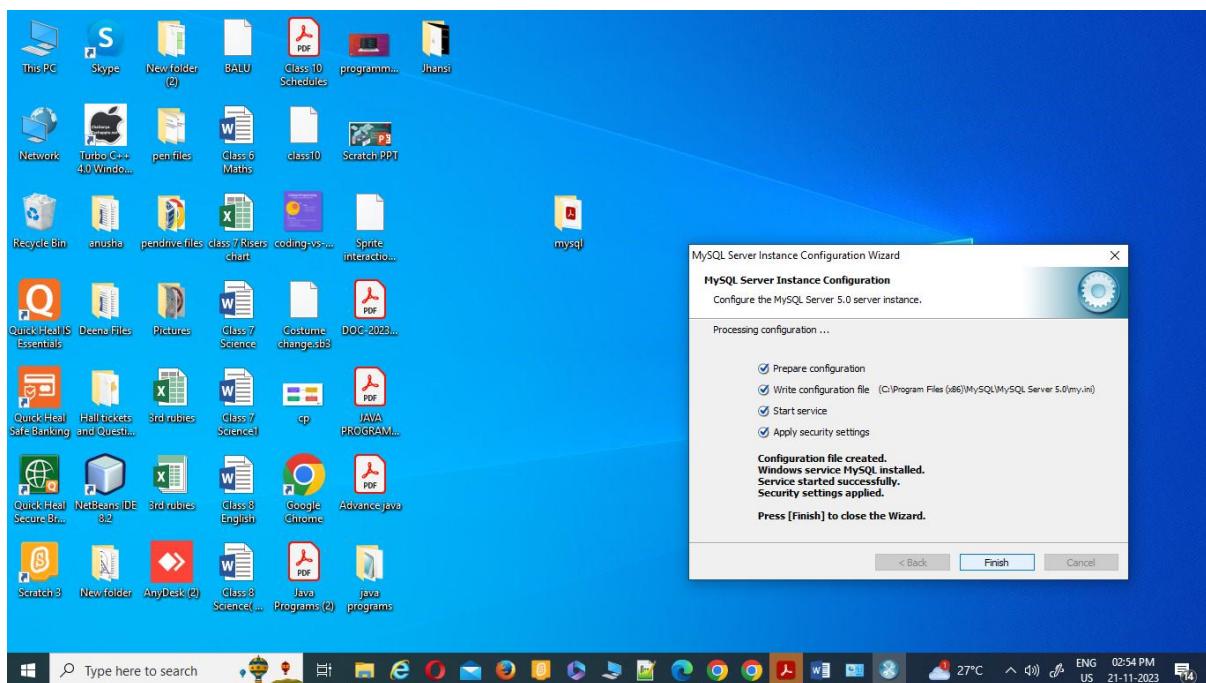
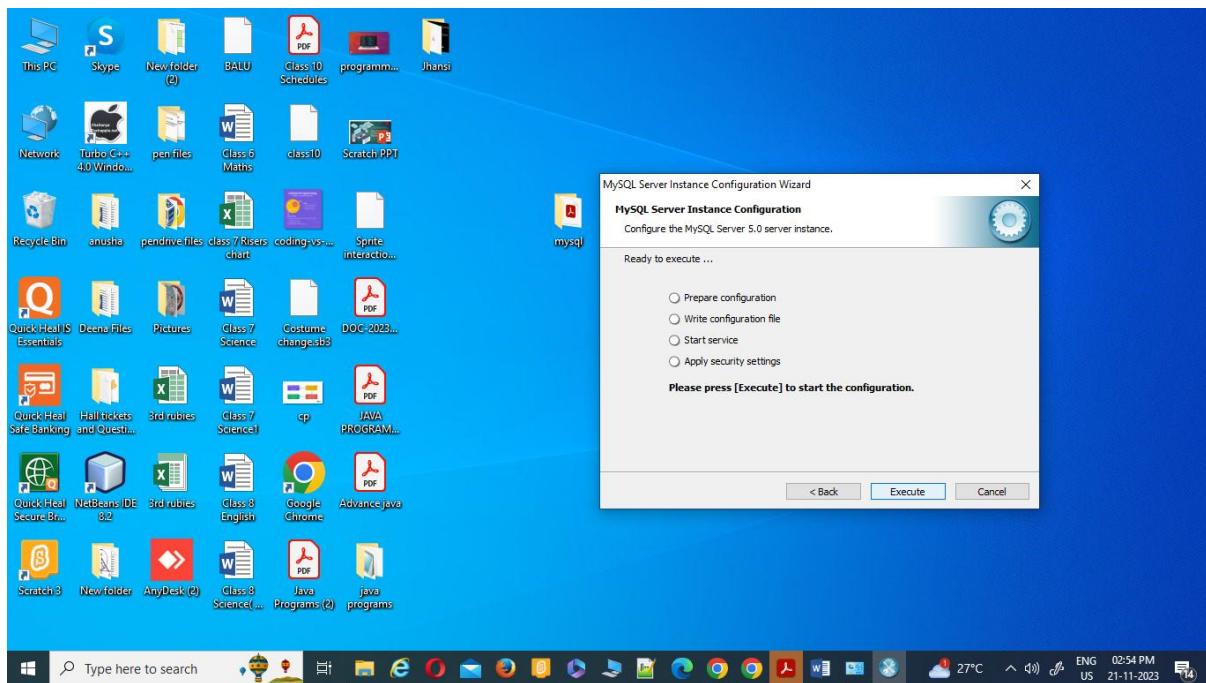




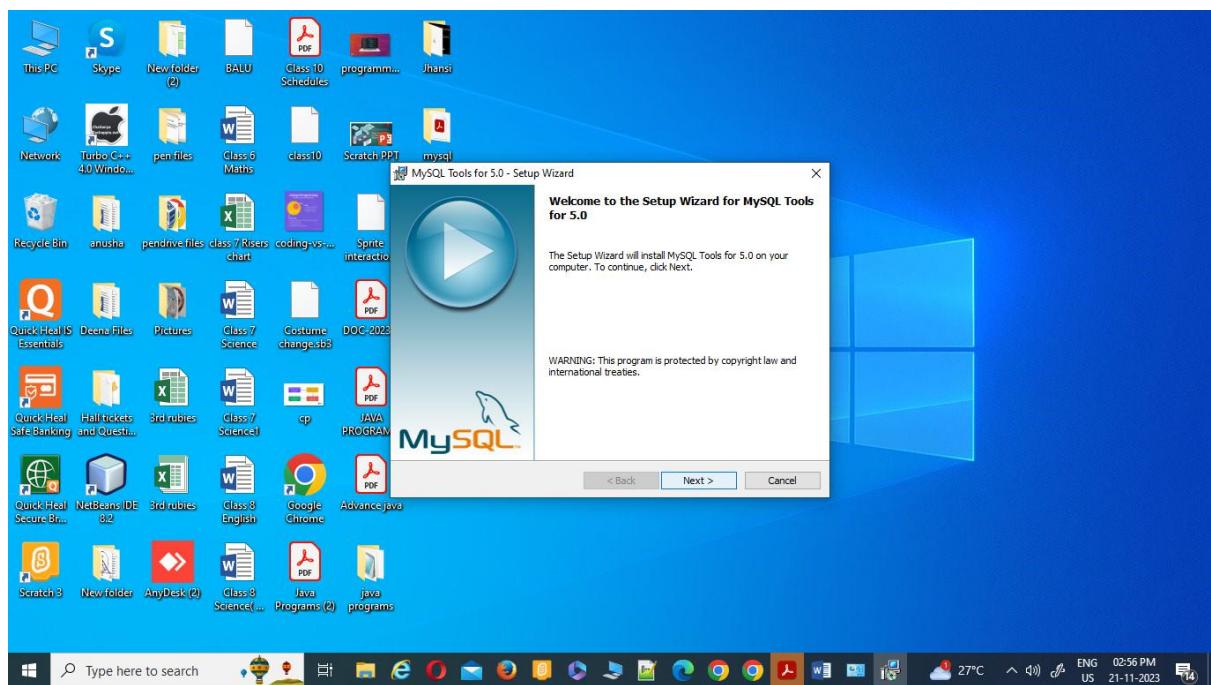
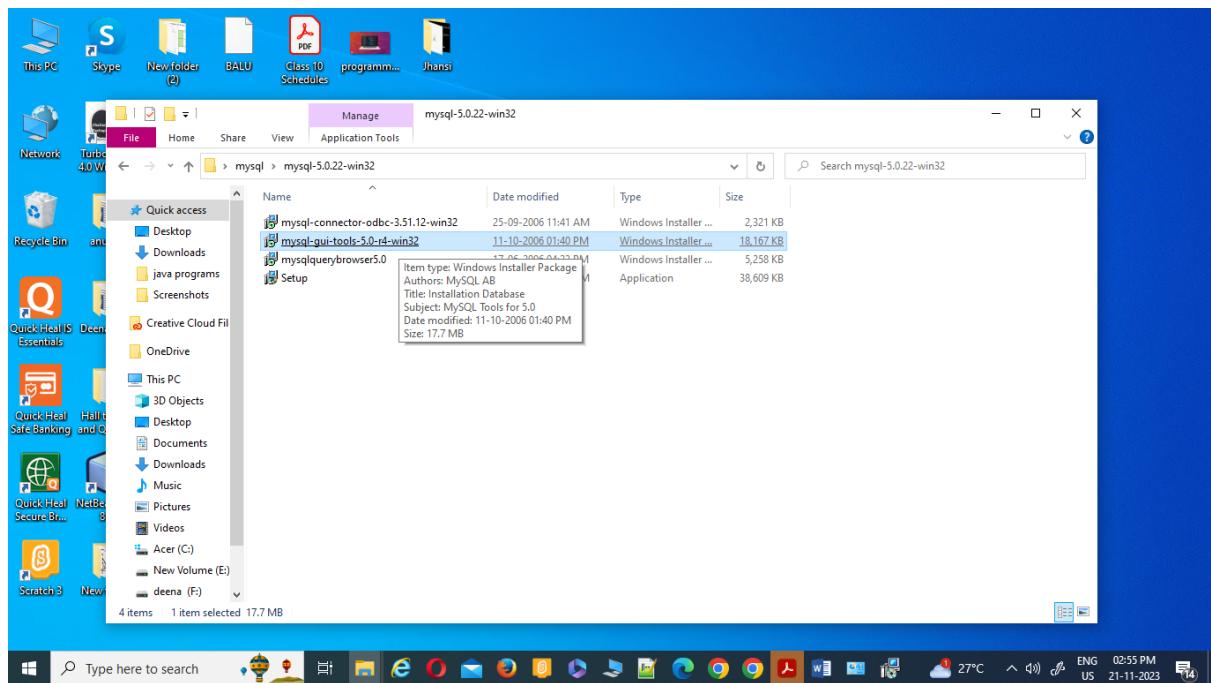


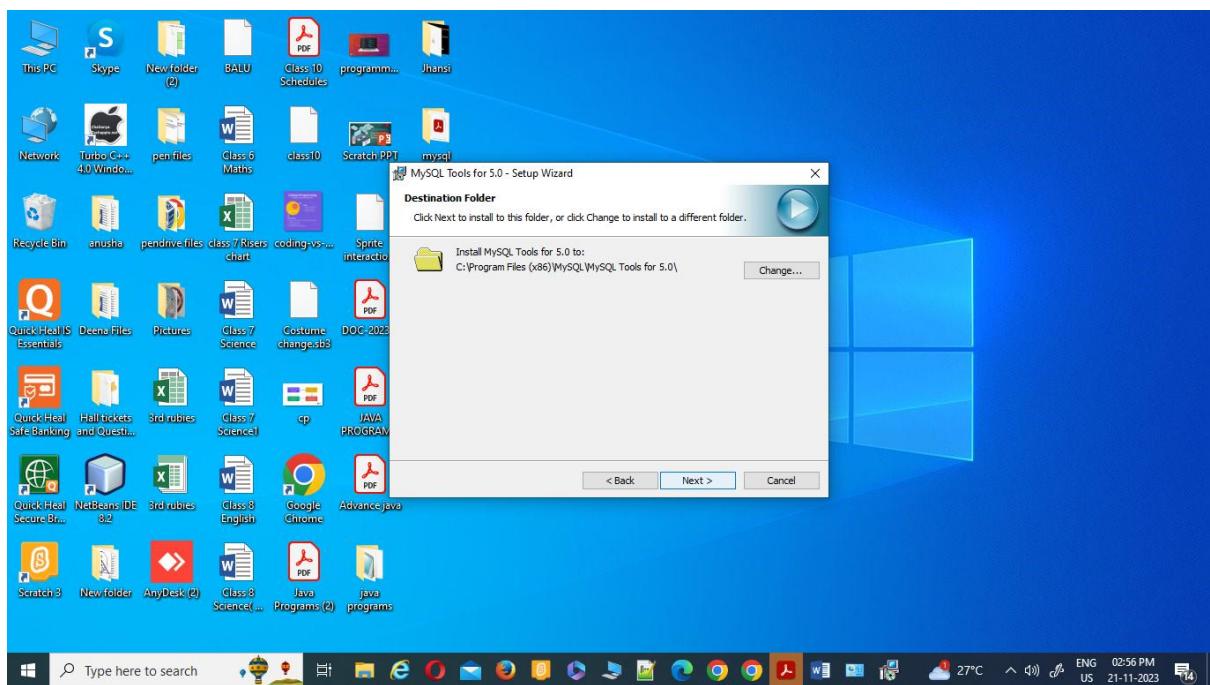
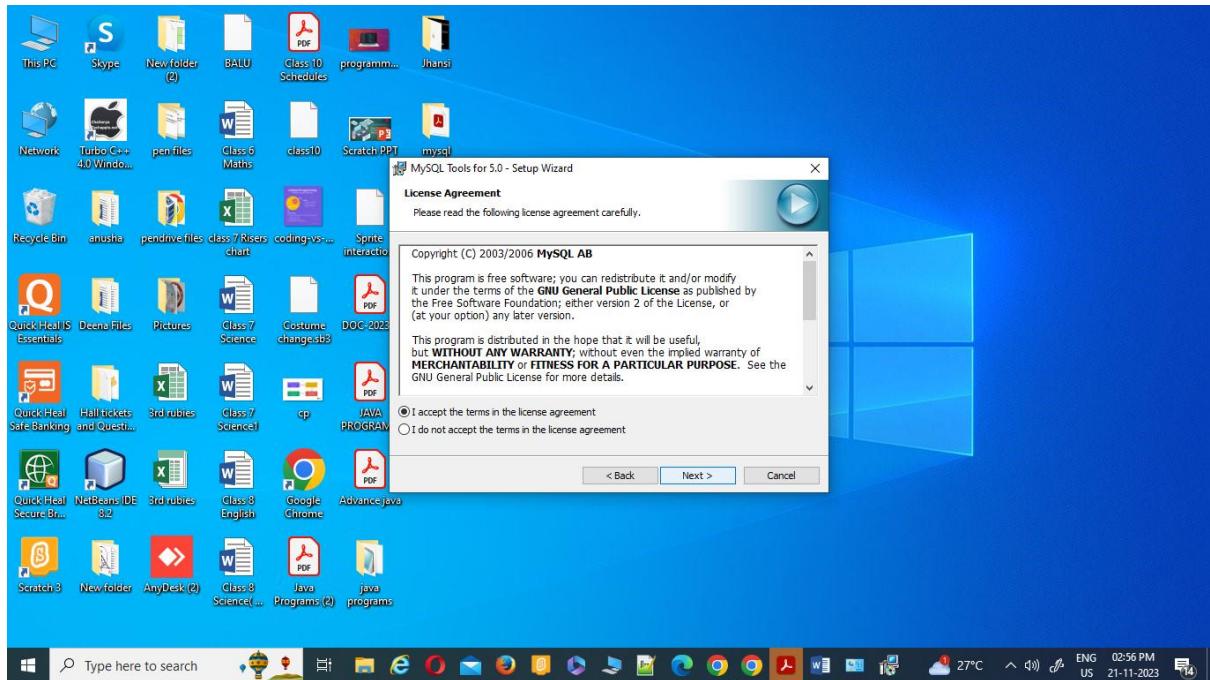
Enter password root
Retype password root

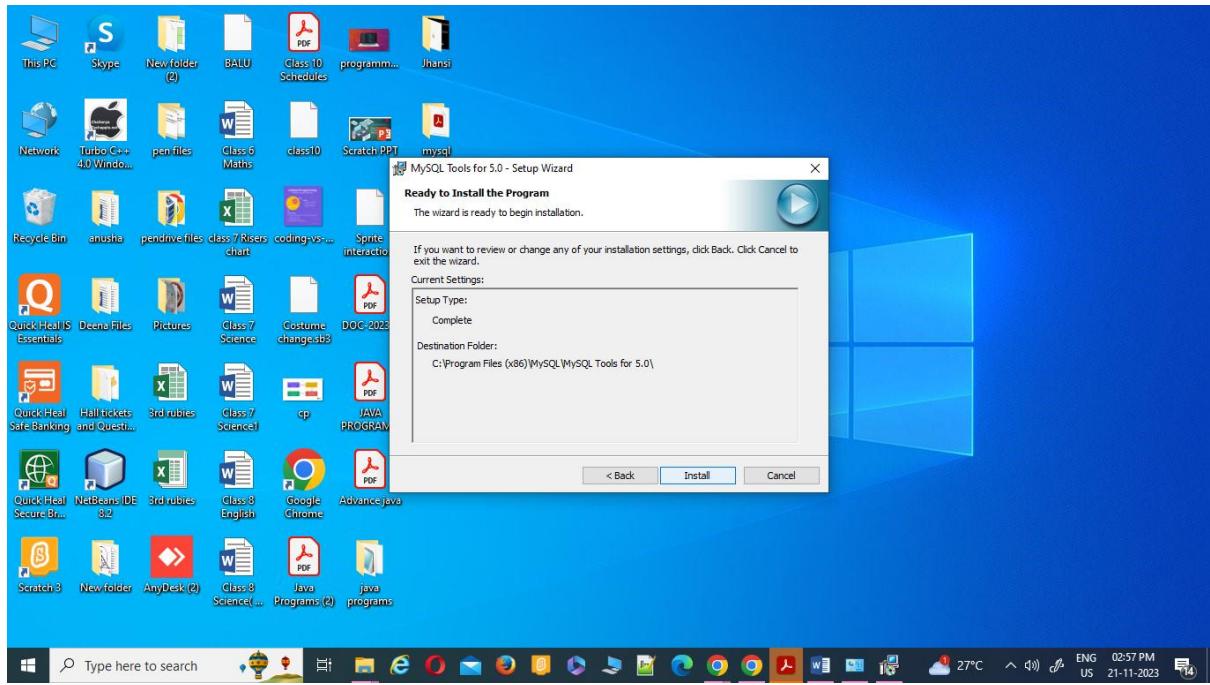
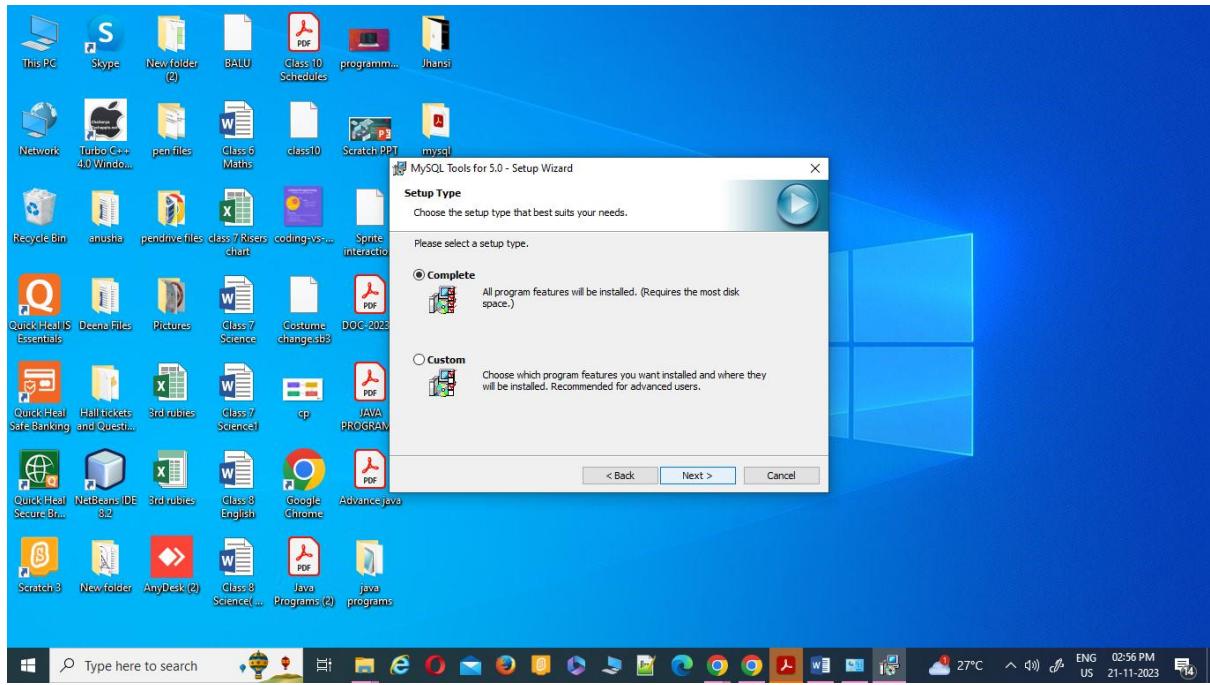


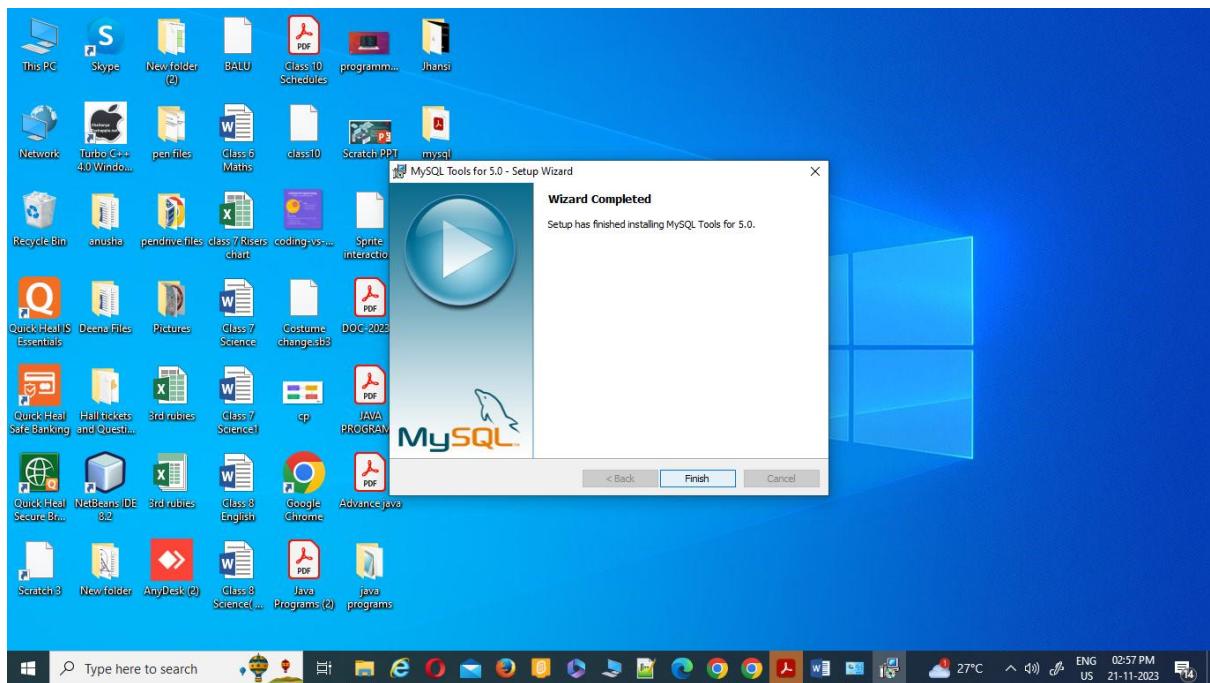
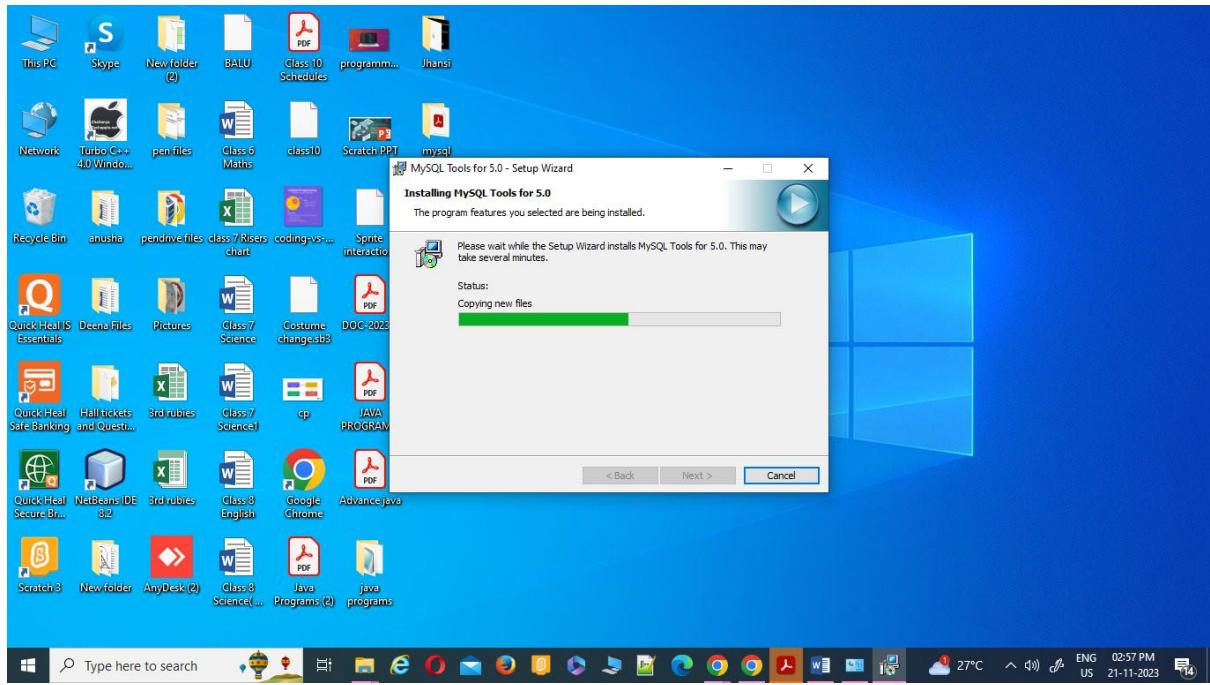


Gui tool installation

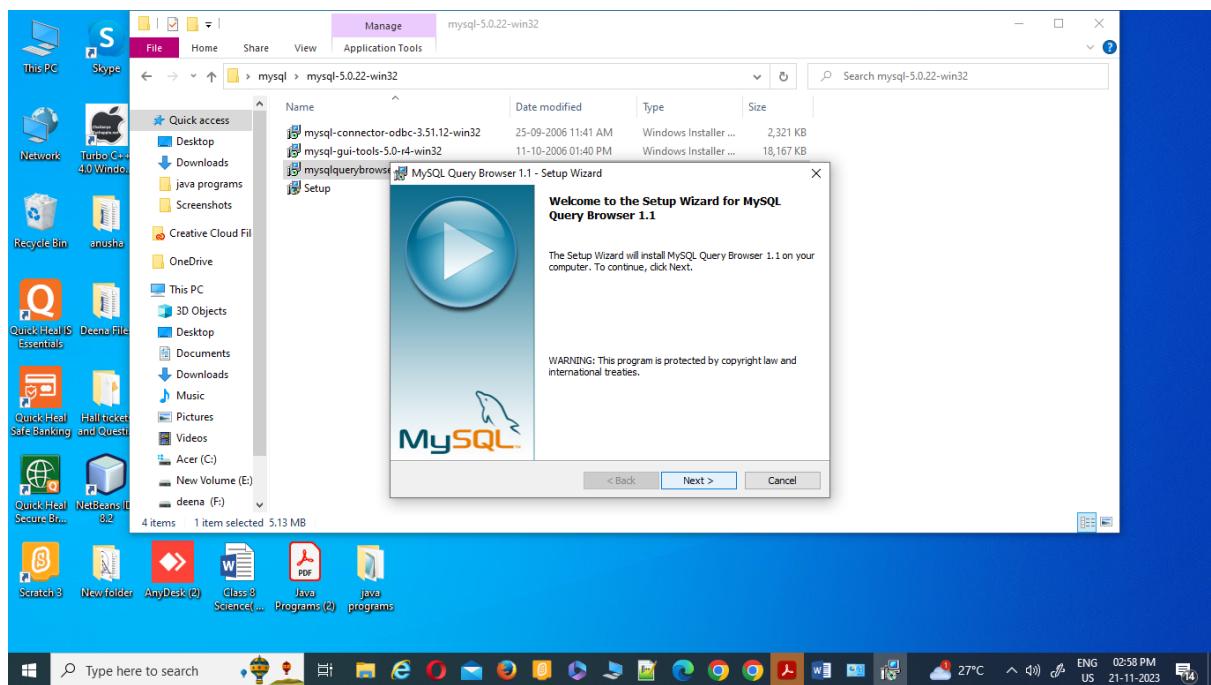
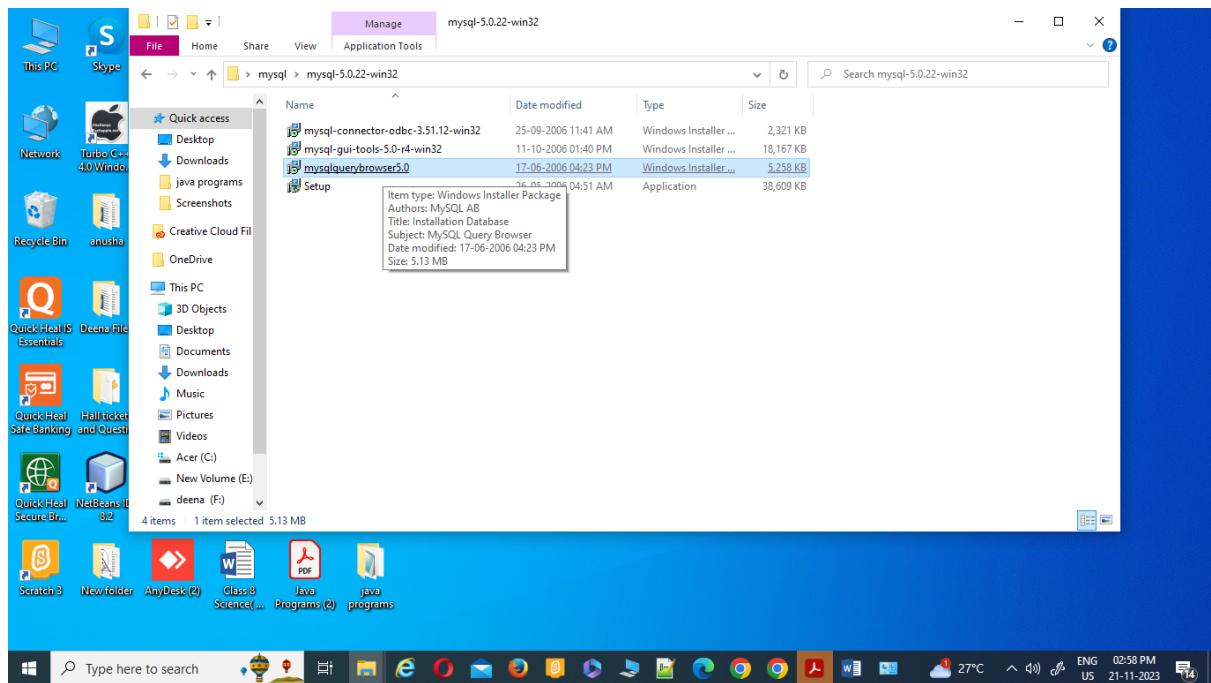


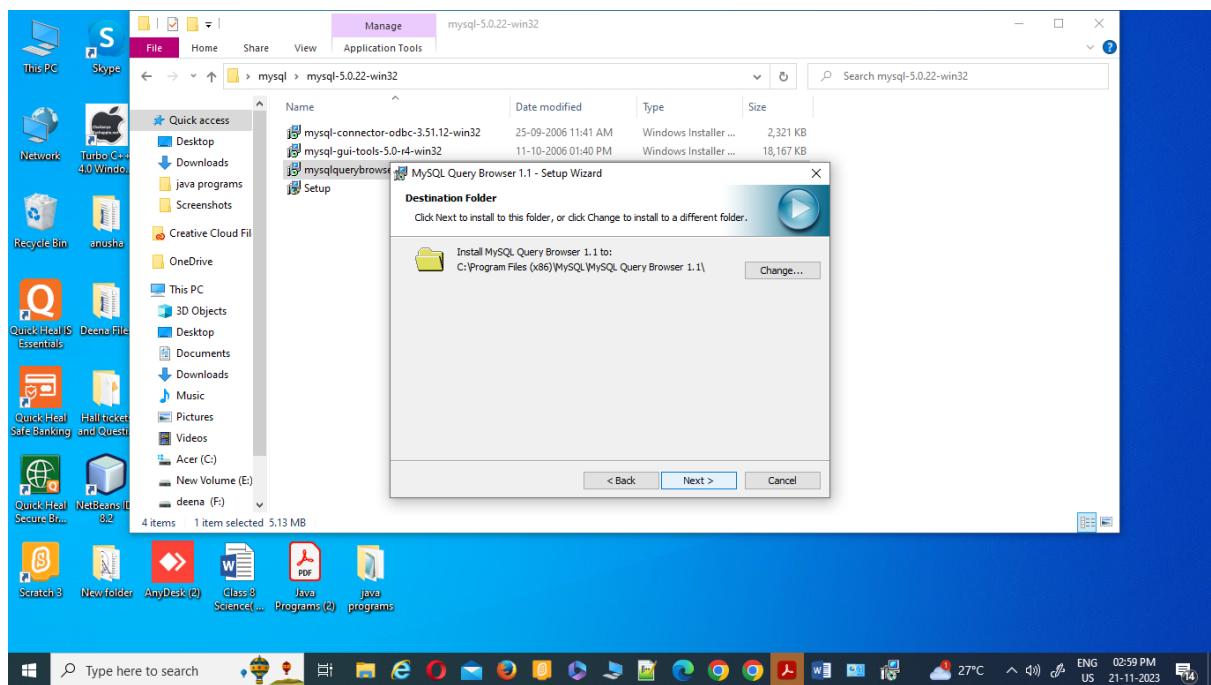
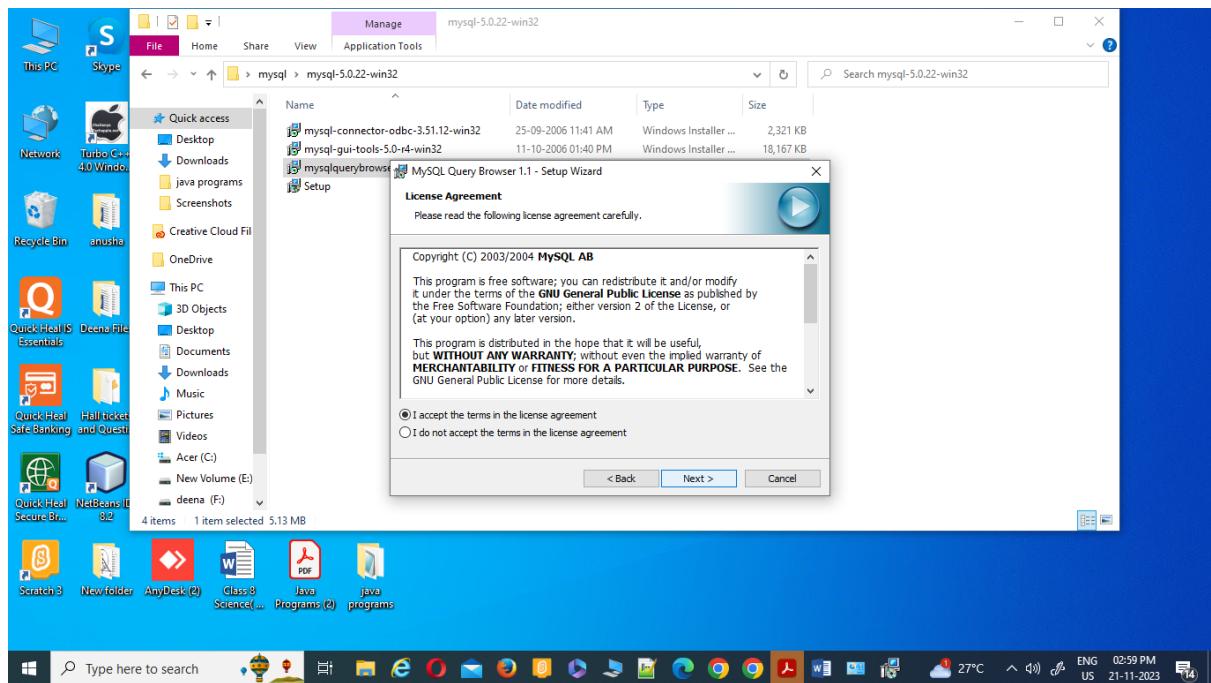


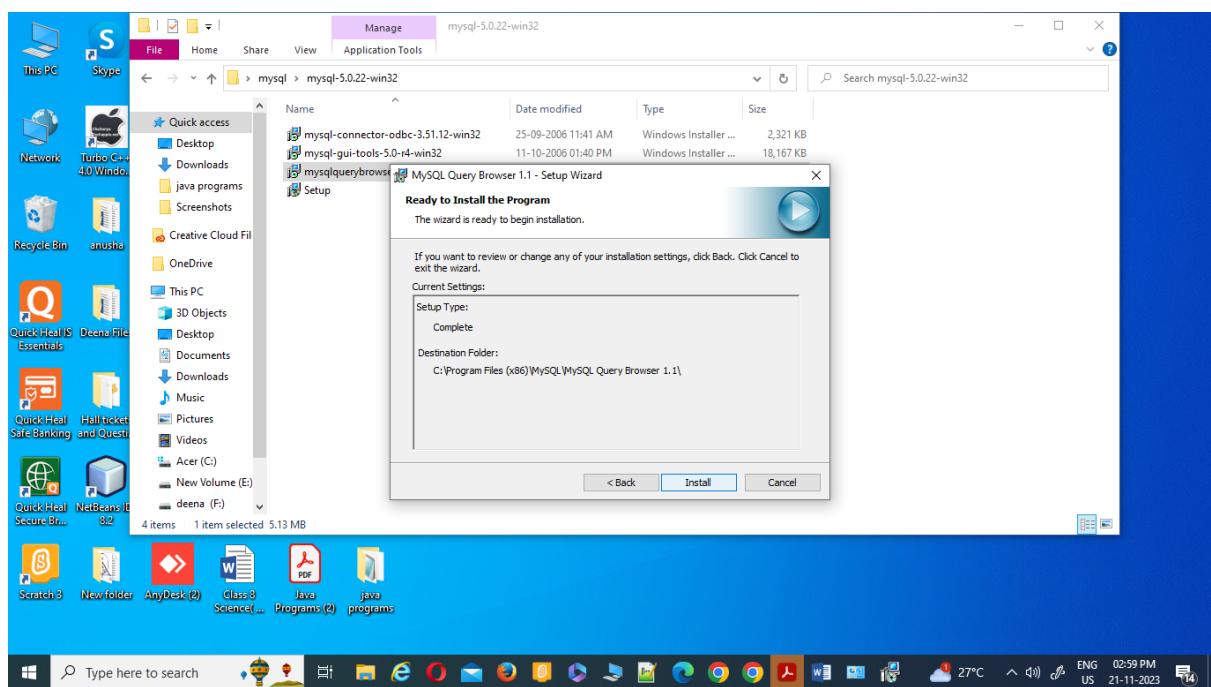
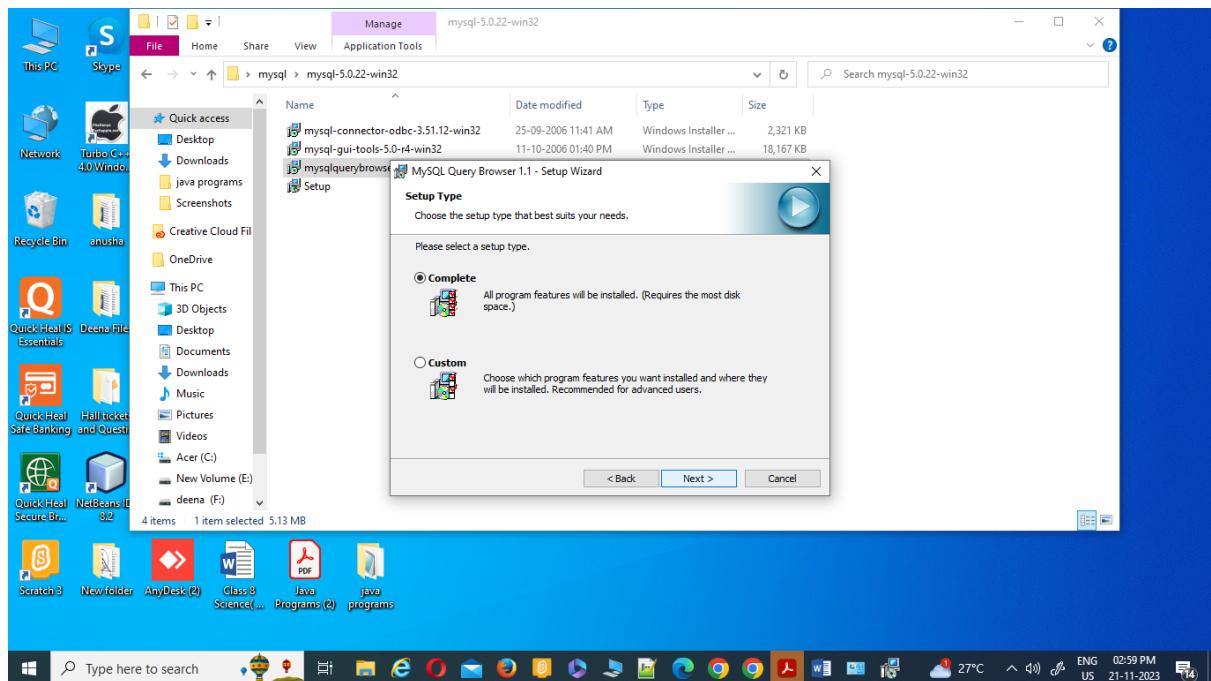


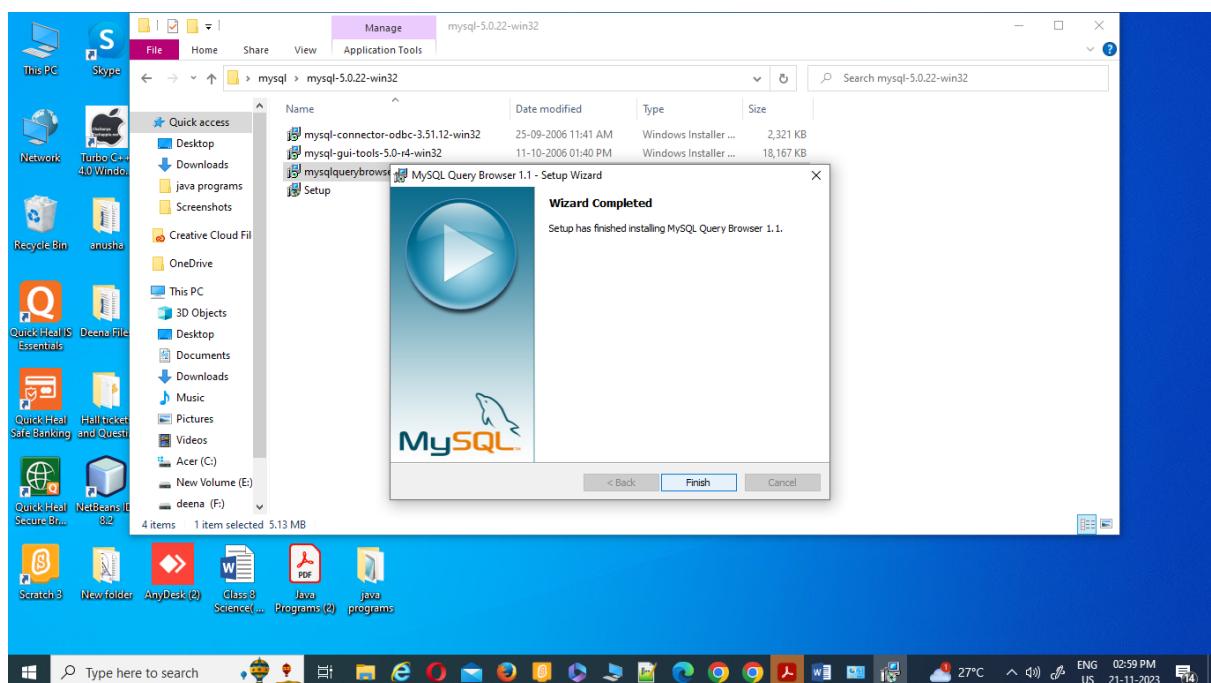
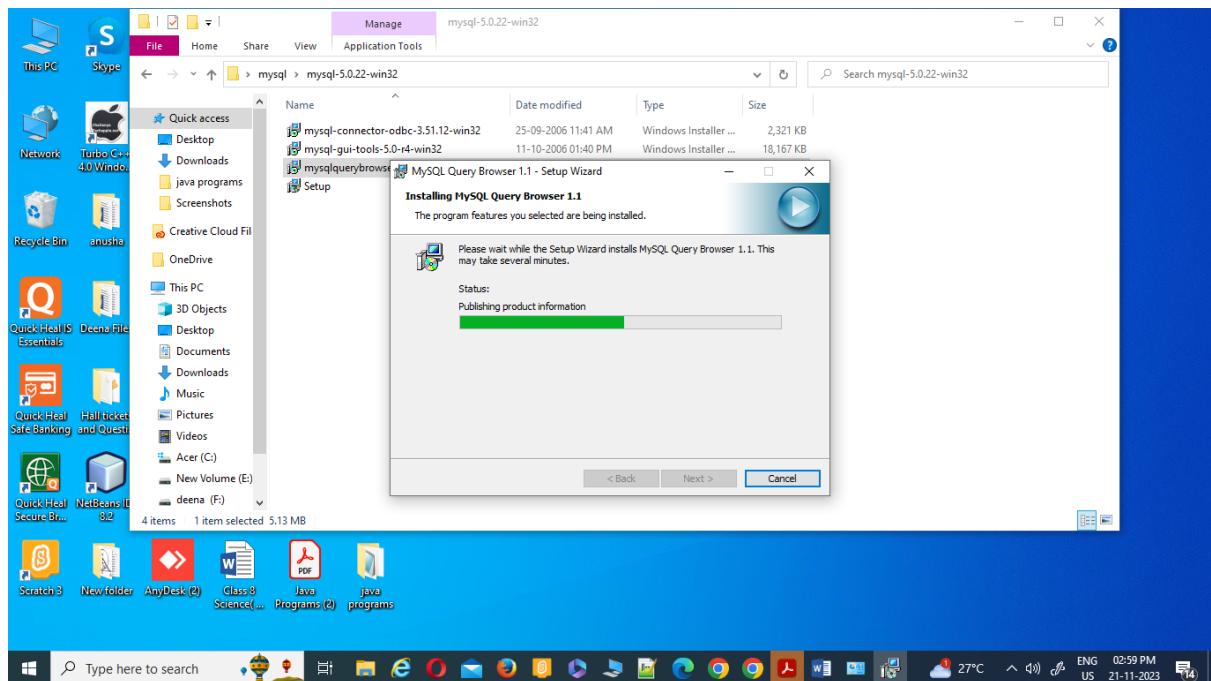


Mysql query browser

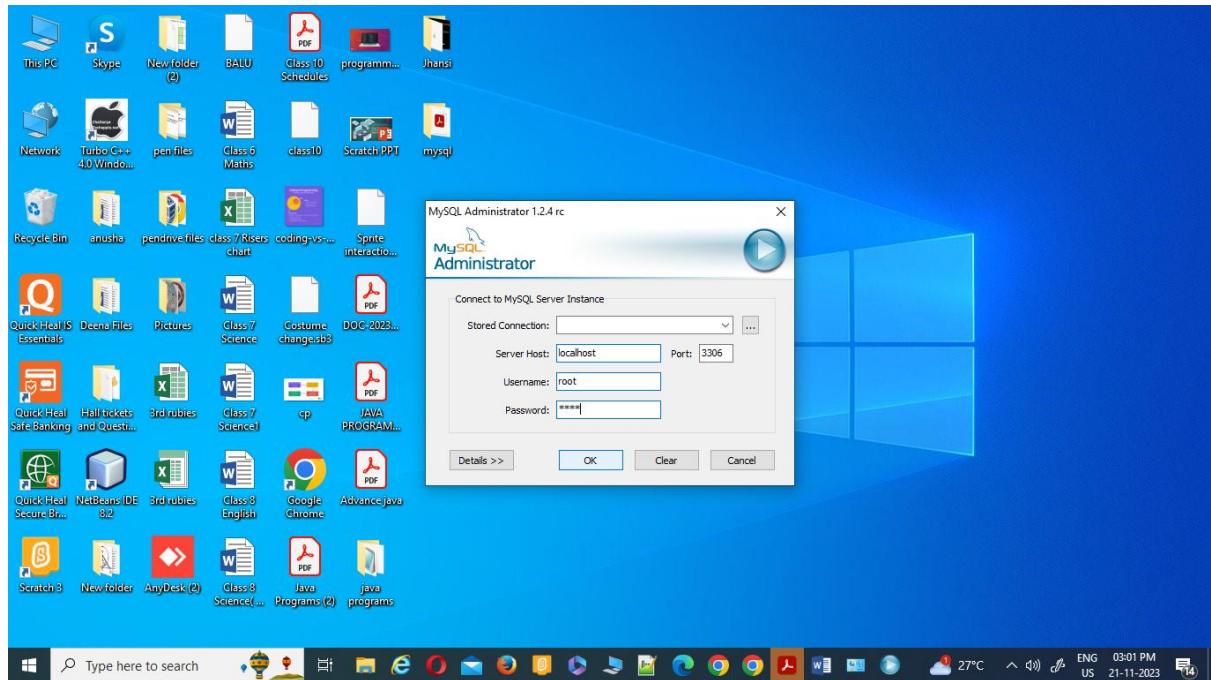






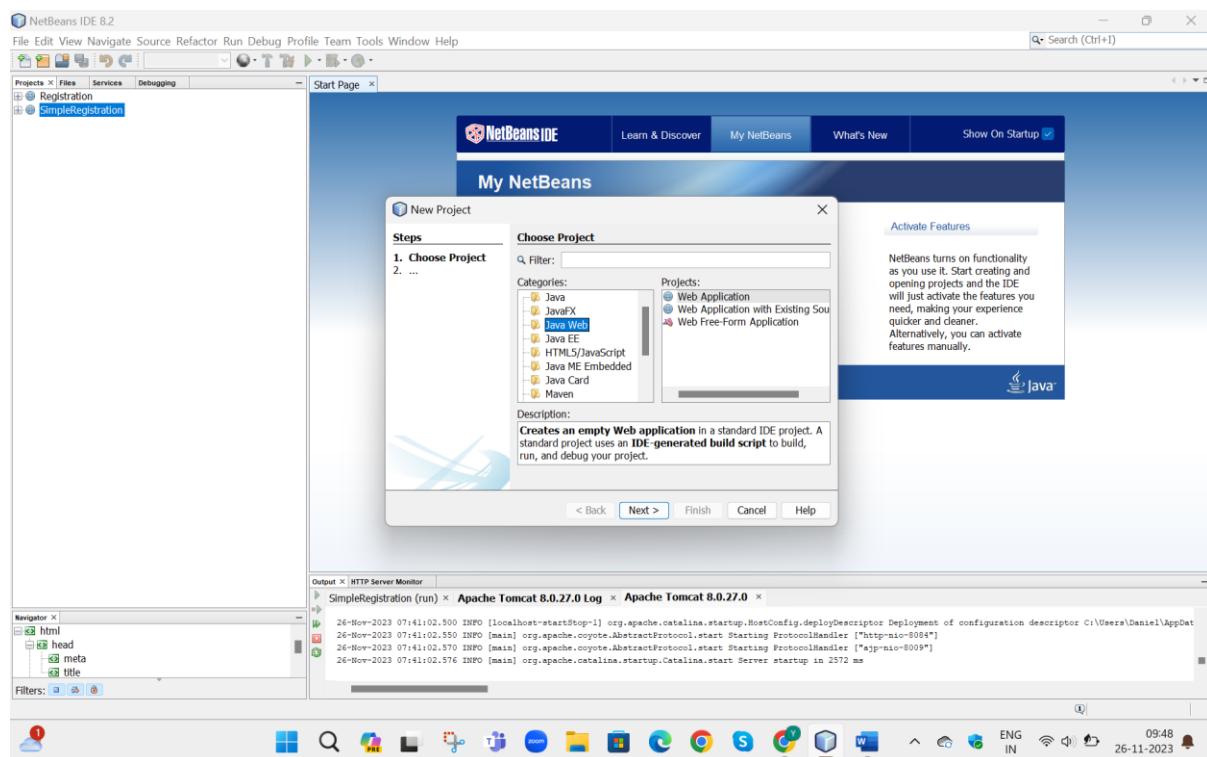
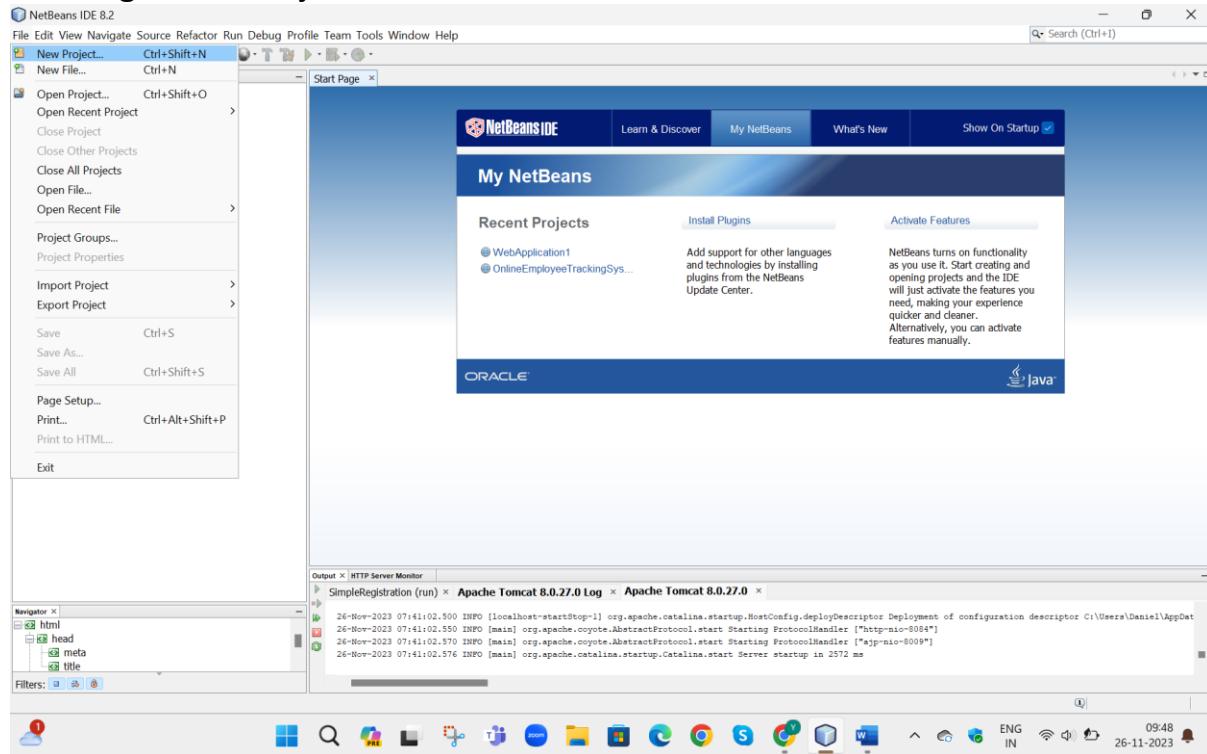


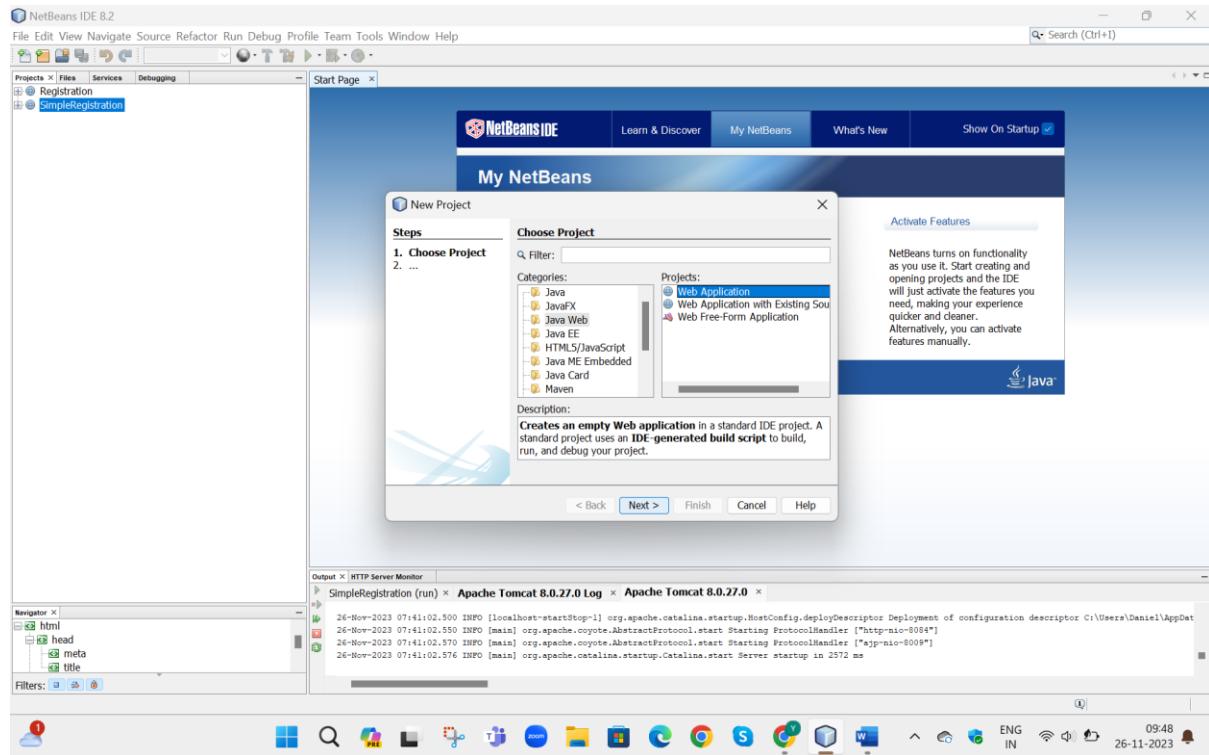
Open mysql administrator



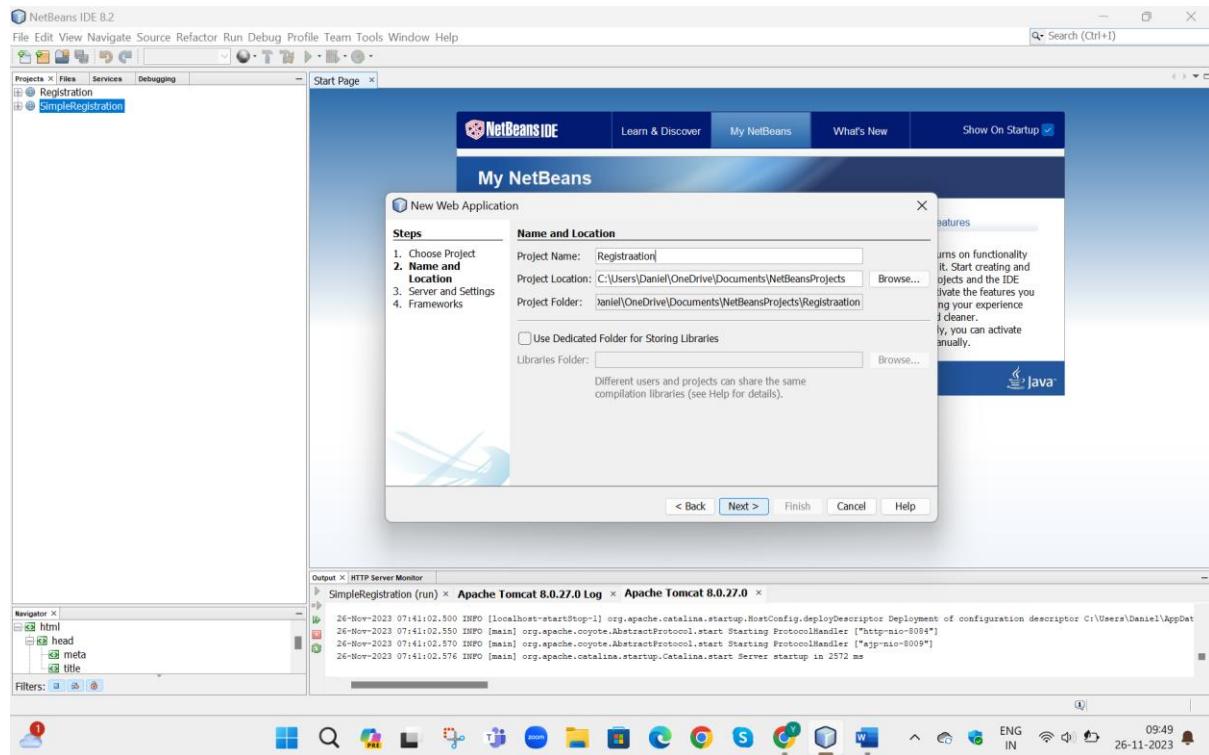
NetBeans Project Creation:

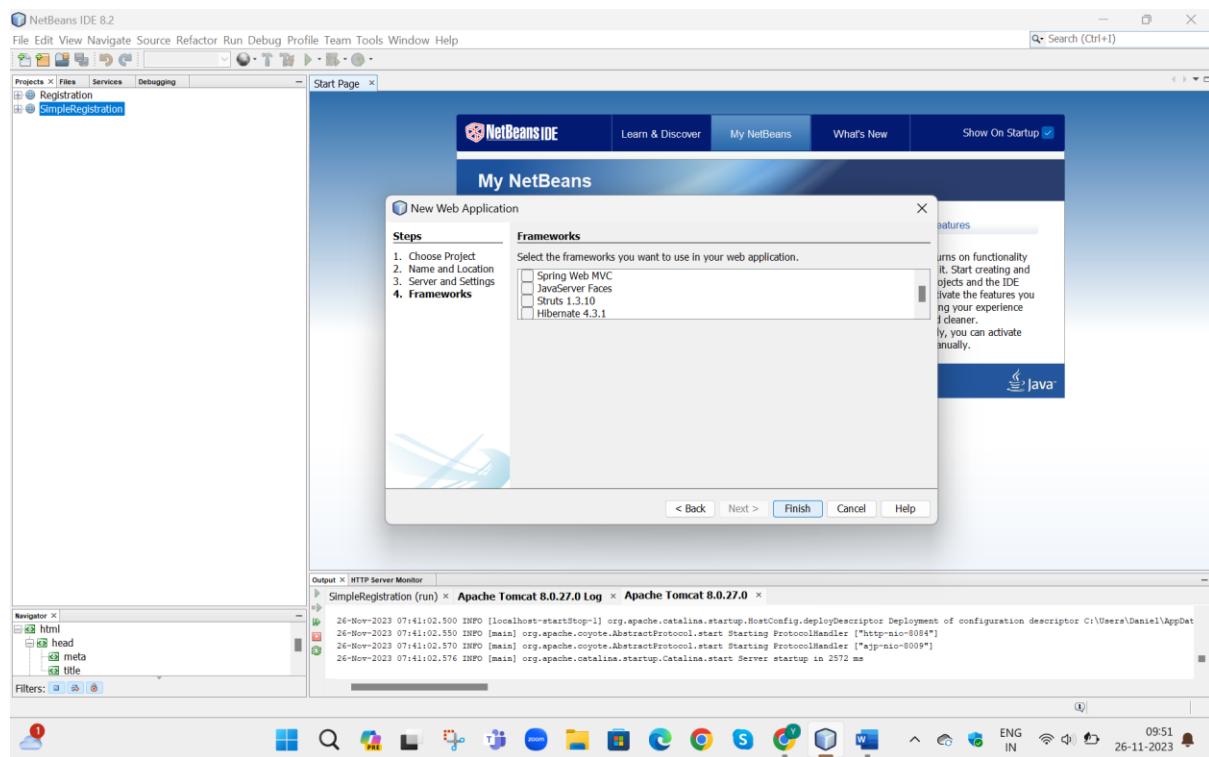
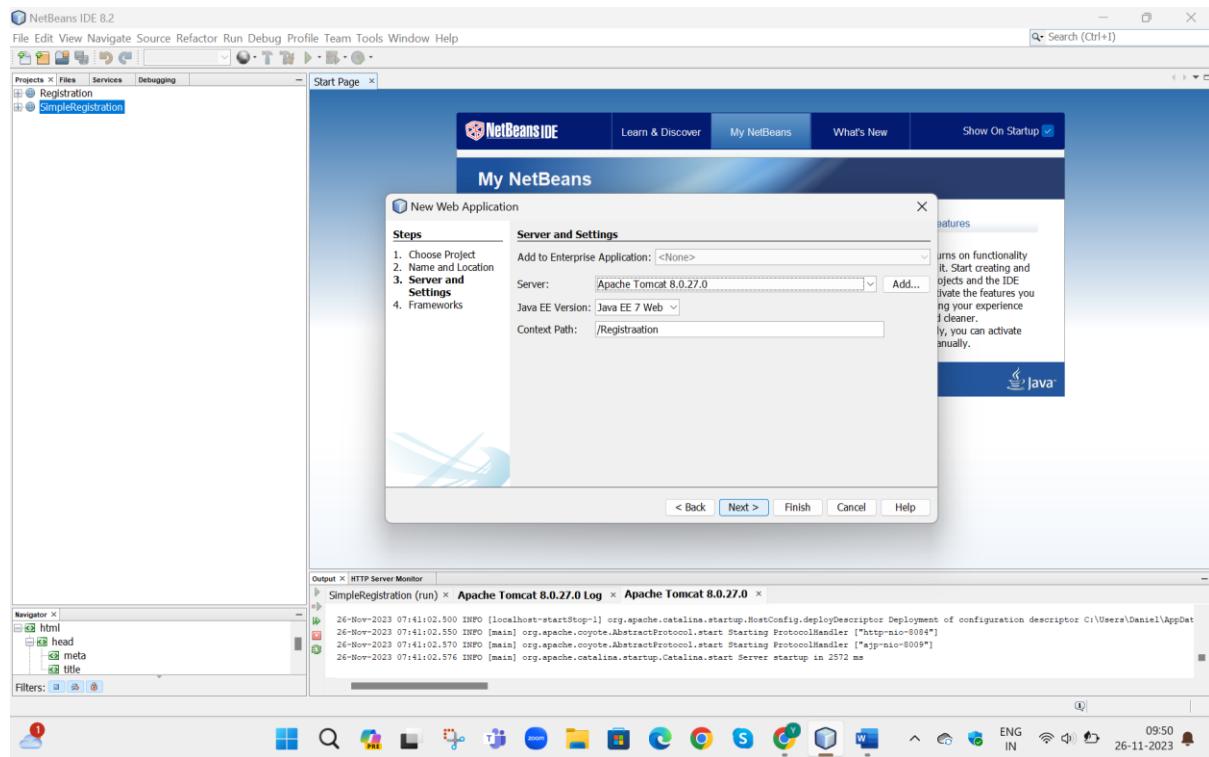
Creating a New Project:





Give a name to your Project (Project name): Registration (Give your Respective Project Name)





Registration - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Registration Registration SimpleRegistration

Start Page index.html

```

<!DOCTYPE html>
<!--
  To change this license header, choose License Headers in Project Properties.
  To change this template file, choose Tools | Templates
  and open the template in the editor.
-->
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div>TODO write content</div>
  </body>
</html>

```

Output X HTTP Server Monitor

SimpleRegistration (run) × Apache Tomcat 8.0.27.0 Log × Apache Tomcat 8.0.27.0

```

26-Nov-2023 07:41:02.500 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDescriptor Deployment of configuration descriptor C:\Users\Daniel\AppData\Local\Temp\Tomcat\8.0.27.0\temp\1\conf\localhost-startStop\index.html
26-Nov-2023 07:41:02.550 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8084"]
26-Nov-2023 07:41:02.570 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
26-Nov-2023 07:41:02.576 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 2572 ms

```

Navigator X

html

17:1 INS

ENG IN 09:51 26-11-2023

Remove the default code and enter the code shared with you as shown below:

index.html

Registration - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Registration Registration SimpleRegistration

Start Page index.html

```

        }
        li
        {
            float: left;
            text-decoration: none;
            list-style: none;
            padding: 10px 10px 10px;
        }

        li a{
            color: white;
            text-decoration: none;
        }

        </style>
    </head>
    <body>
        <div class = "container">

            <nav>
                <ul>
                    <li><a href = "index.html" >HOME</a></li>
                    <li><a href = "index.html" >ABOUT</a></li>
                    <li><a href = "index.html" >CONTACT</a></li>
                    <li style="float: right;"><a href = "register.jsp" >REGISTER</a></li>
                    <li style="float: right;"><a href = "login.jsp" >LOGIN</a></li>
                    <li><a href = "index.html" >OUR SERVICES</a></li>
                </ul>
            </nav>
        </div>
    </body>

```

Output X HTTP Server Monitor

SimpleRegistration (run) × Apache Tomcat 8.0.27.0 Log × Apache Tomcat 8.0.27.0

```

26-Nov-2023 07:41:02.500 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDescriptor Deployment of configuration descriptor C:\Users\Daniel\AppData\Local\Temp\Tomcat\8.0.27.0\temp\1\conf\localhost-startStop\index.html
26-Nov-2023 07:41:02.550 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8084"]
26-Nov-2023 07:41:02.570 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
26-Nov-2023 07:41:02.576 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 2572 ms

```

Navigator X

css

html

head

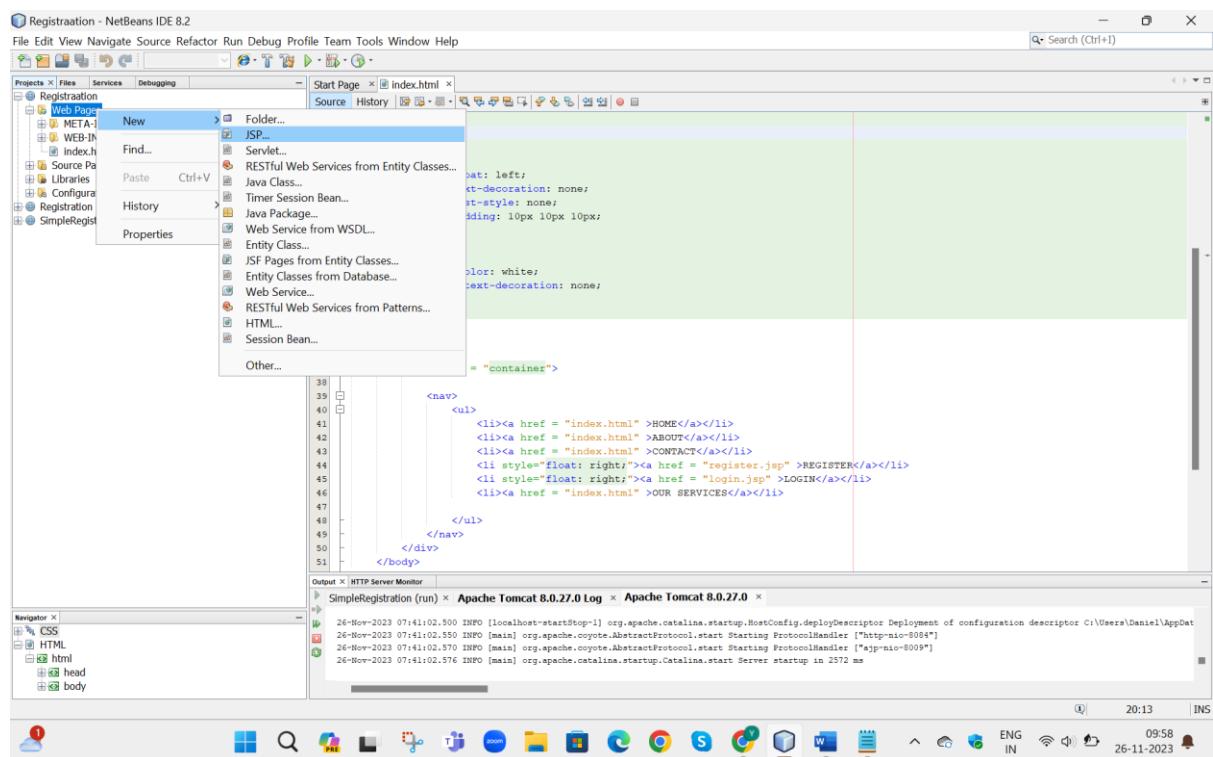
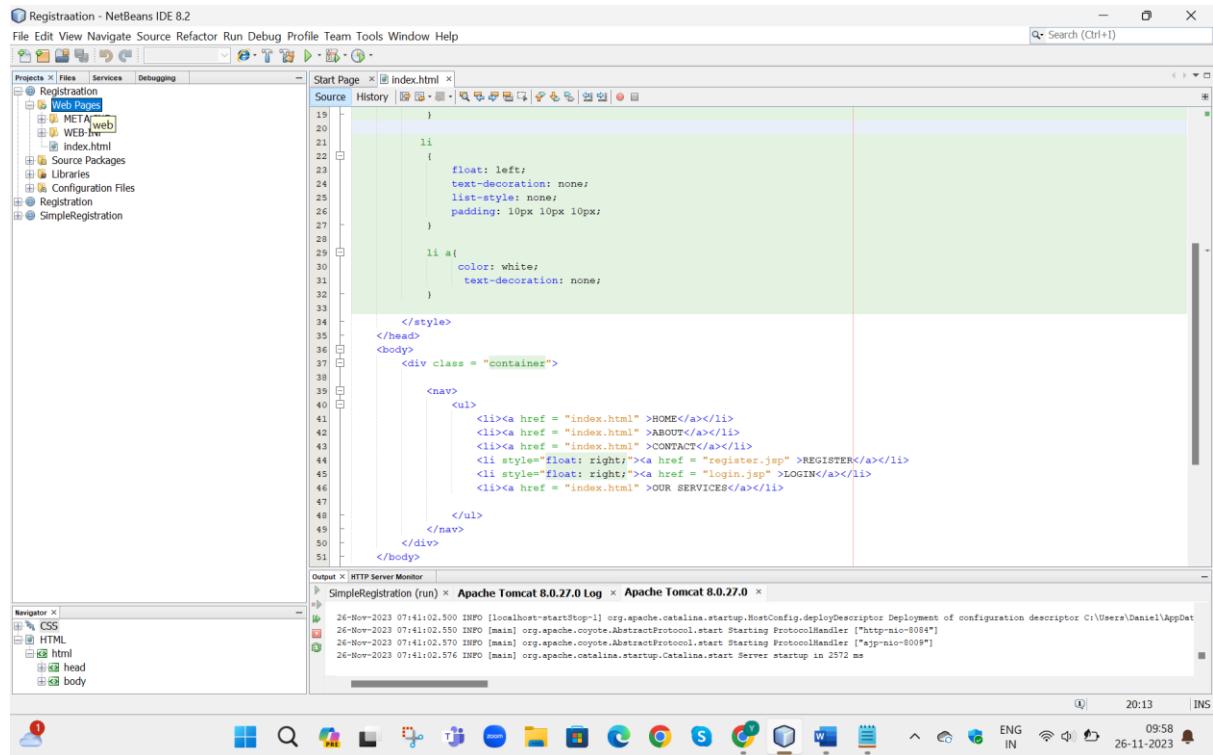
body

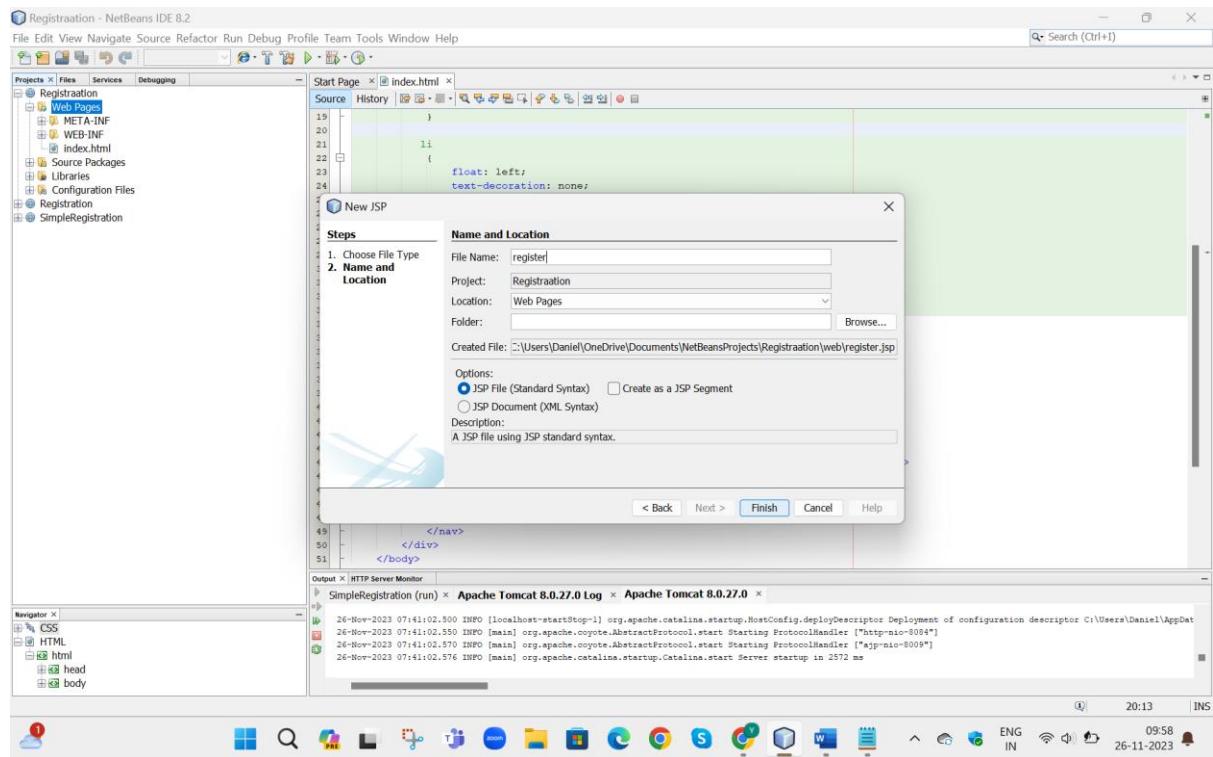
20:13 INS

ENG IN 09:57 26-11-2023

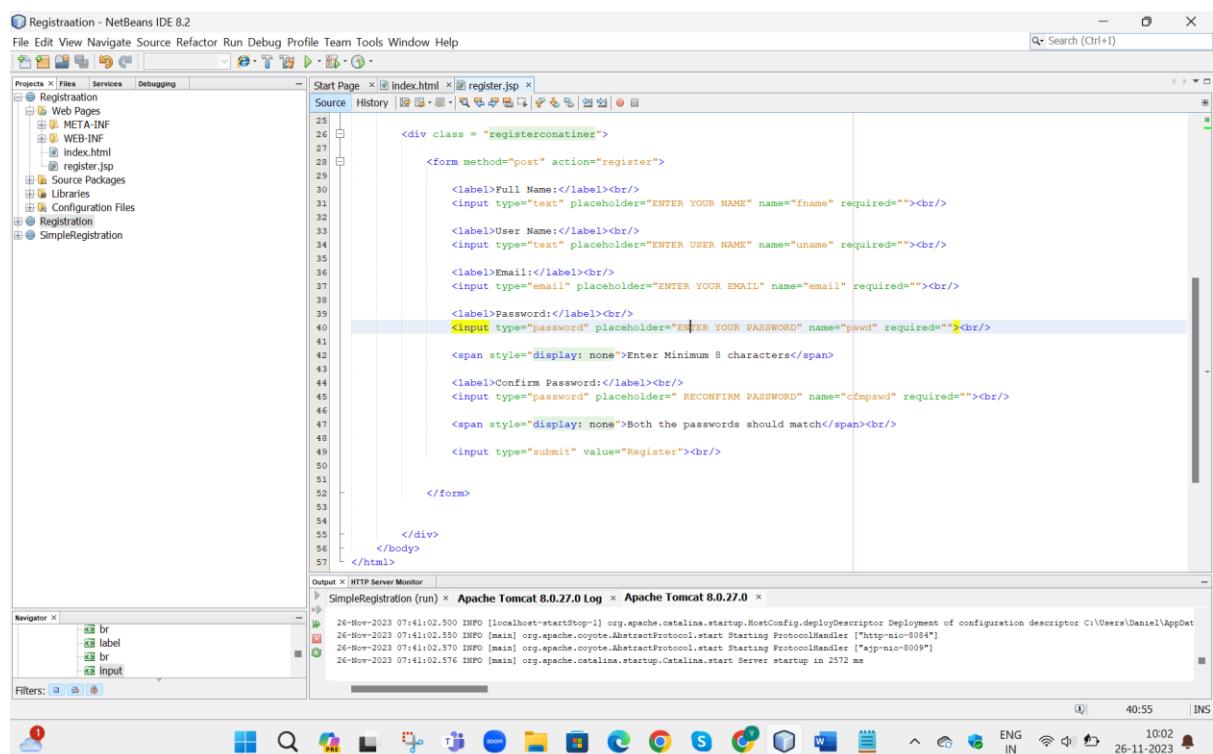
Creating jsp pages:

Right click on Web Pages -> New -> JSP as shown below

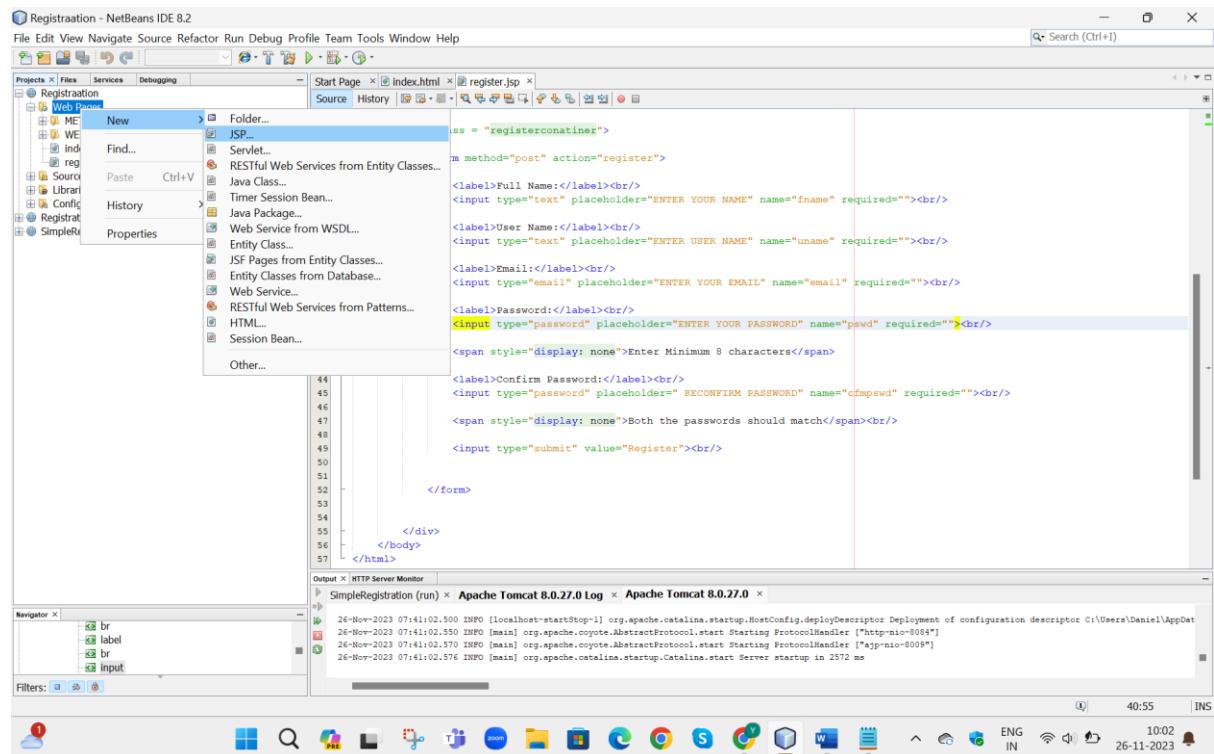




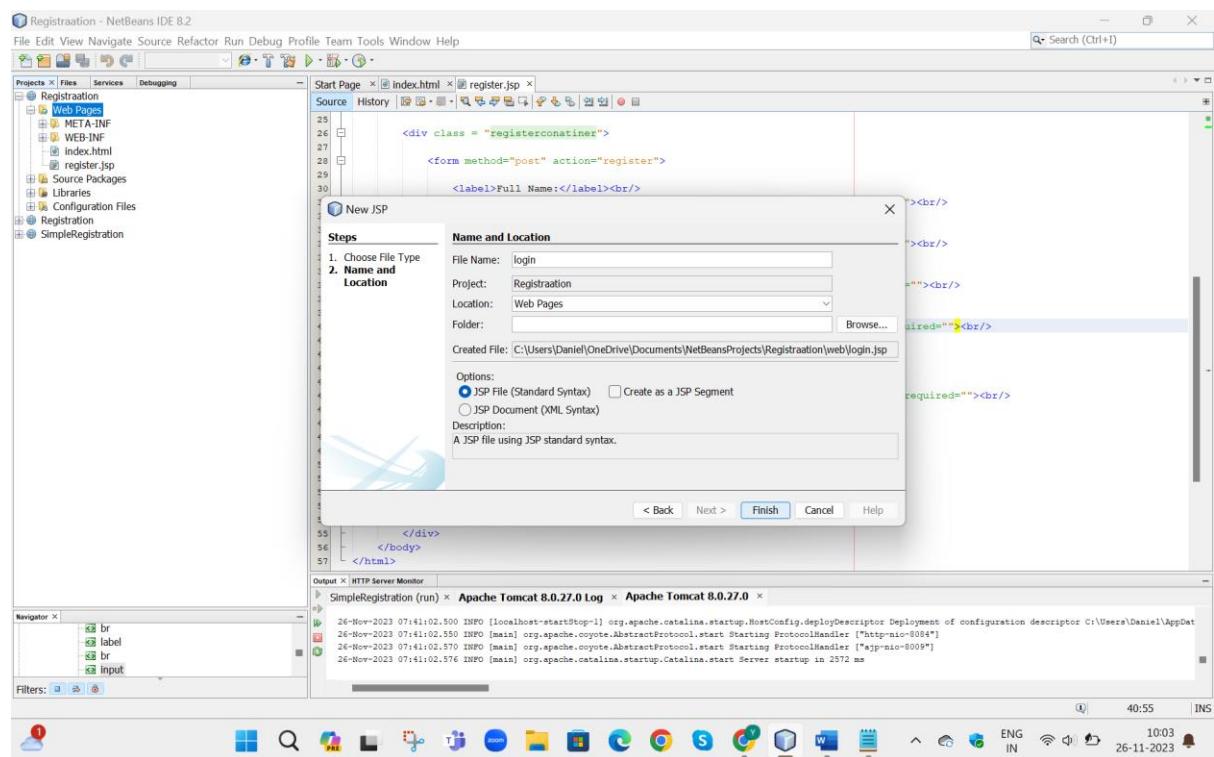
enter filename as “**register**” and hit **Finish** button.
 Remove the default code and enter the code shared with you given in word document for “**register.jsp**” as shown below:



Again, create another file with name “**login.jsp**” as shown below:



With file name as “**login.jsp**” as shown below:



Hit **Finish** button and enter the code for it as shown below:

Registration - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects X Files Services Debugging

Start Page x index.html x register.jsp x login.jsp x

Source History

Document : login
Created on : 16 Nov, 2023, 1:01:26 PM
Author : Daniel

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body align="center">

        <form method="post" action="login">
            <label>User Name:</label><br/>
            <input type="text" placeholder="ENTER USER NAME" name="uname" required=""><br/>

            <label>Password:</label><br/>
            <input type="password" placeholder="ENTER YOUR PASSWORD" name="pswd" required=""><br/><br/>

            <input type="submit" value="Login"><br/>
        </form>
    </body>
</html>

```

Output x HTTP Server Monitor

SimpleRegistration (run) x Apache Tomcat 8.0.27.0 Log x Apache Tomcat 8.0.27.0 x

```

26-Nov-2023 07:41:02.500 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDescriptor Deployment of configuration descriptor C:\Users\Daniel\AppData\Local\Temp\Tomcat\conf\localhost-startStop\1\descriptor.xml completed
26-Nov-2023 07:41:02.550 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8084"]
26-Nov-2023 07:41:02.570 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
26-Nov-2023 07:41:02.576 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 2572 ms

```

Navigator X

Filters: a b c

13:12 INS

10:05 ENG IN 26-11-2023

Again, with file name as “error.jsp” as shown below:

Registration - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects X Files Services Debugging

Start Page x index.html x register.jsp x login.jsp x

Source History

New

- Web Pages
 - MET
 - WEE
 - Inde
 - logi
 - regi
 - Source
 - Librar
 - Confi
 - Registr
 - SimpleRegistration
- Folder...
- JSP...
- Servlet...
- RESTful Web Services from Entity Classes...
- Java Class...
- Timer Session Bean...
- Java Package...
- Web Service from WSDL...
- Entity Class...
- JSF Pages from Entity Classes...
- Entity Classes from Database...
- Web Service...
- RESTful Web Services from Patterns...
- HTML...
- Session Bean...
- Other...

login
16 Nov, 2023, 1:01:26 PM
Daniel

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body align="center">

        <form method="post" action="login">
            <label>User Name:</label><br/>
            <input type="text" placeholder="ENTER USER NAME" name="uname" required=""><br/>

            <label>Password:</label><br/>
            <input type="password" placeholder="ENTER YOUR PASSWORD" name="pswd" required=""><br/><br/>

            <input type="submit" value="Login"><br/>
        </form>
    </body>
</html>

```

Output x HTTP Server Monitor

SimpleRegistration (run) x Apache Tomcat 8.0.27.0 Log x Apache Tomcat 8.0.27.0 x

```

26-Nov-2023 07:41:02.500 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDescriptor Deployment of configuration descriptor C:\Users\Daniel\AppData\Local\Temp\Tomcat\conf\localhost-startStop\1\descriptor.xml completed
26-Nov-2023 07:41:02.550 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8084"]
26-Nov-2023 07:41:02.570 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
26-Nov-2023 07:41:02.576 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 2572 ms

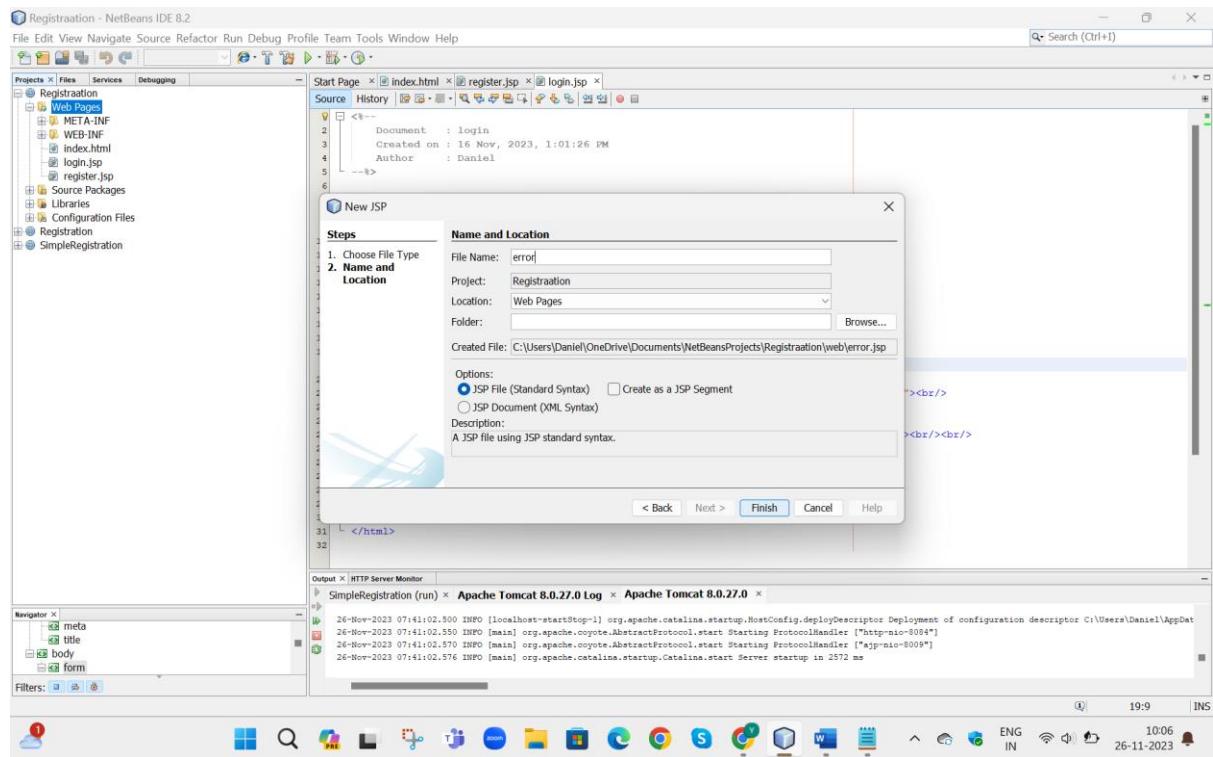
```

Navigator X

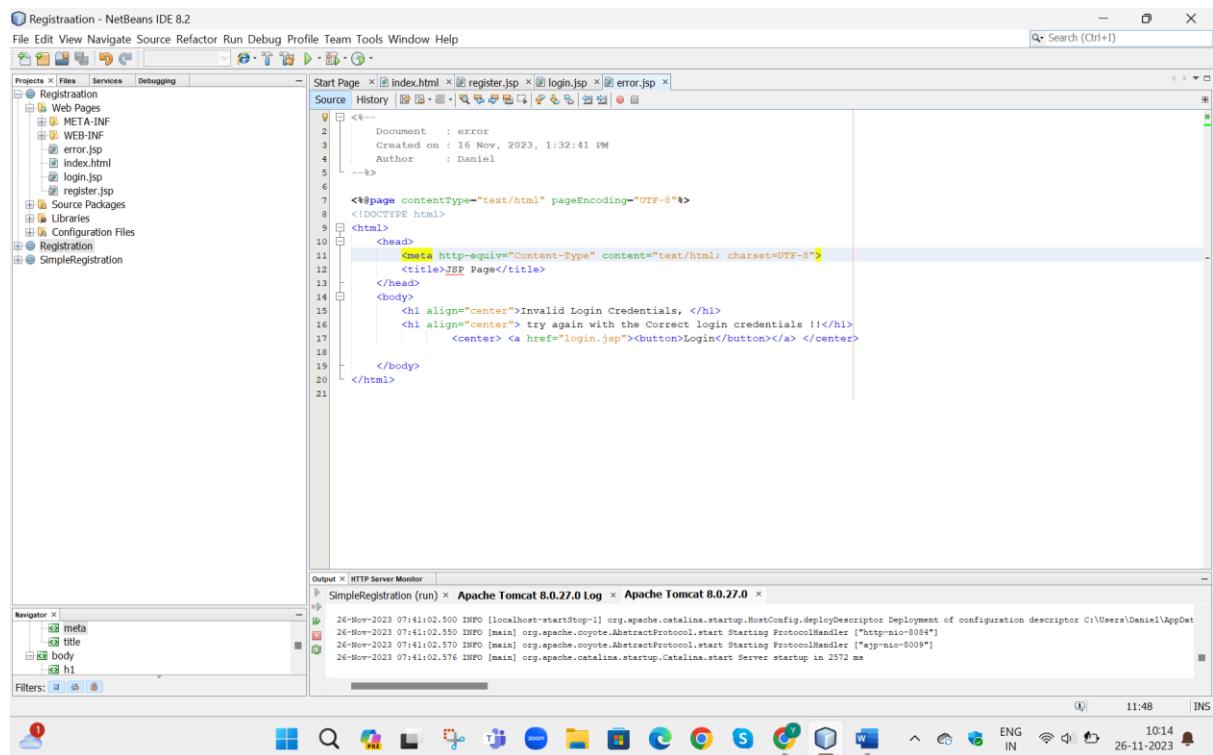
Filters: a b c

19:9 INS

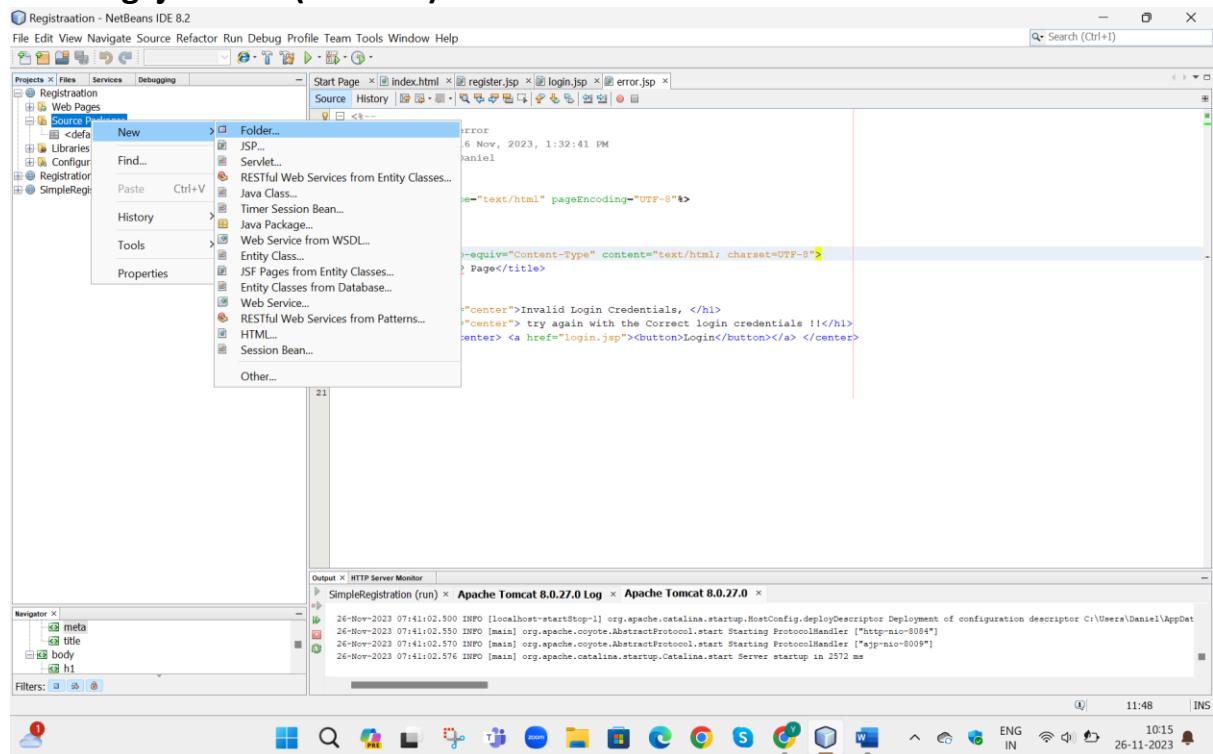
10:05 ENG IN 26-11-2023



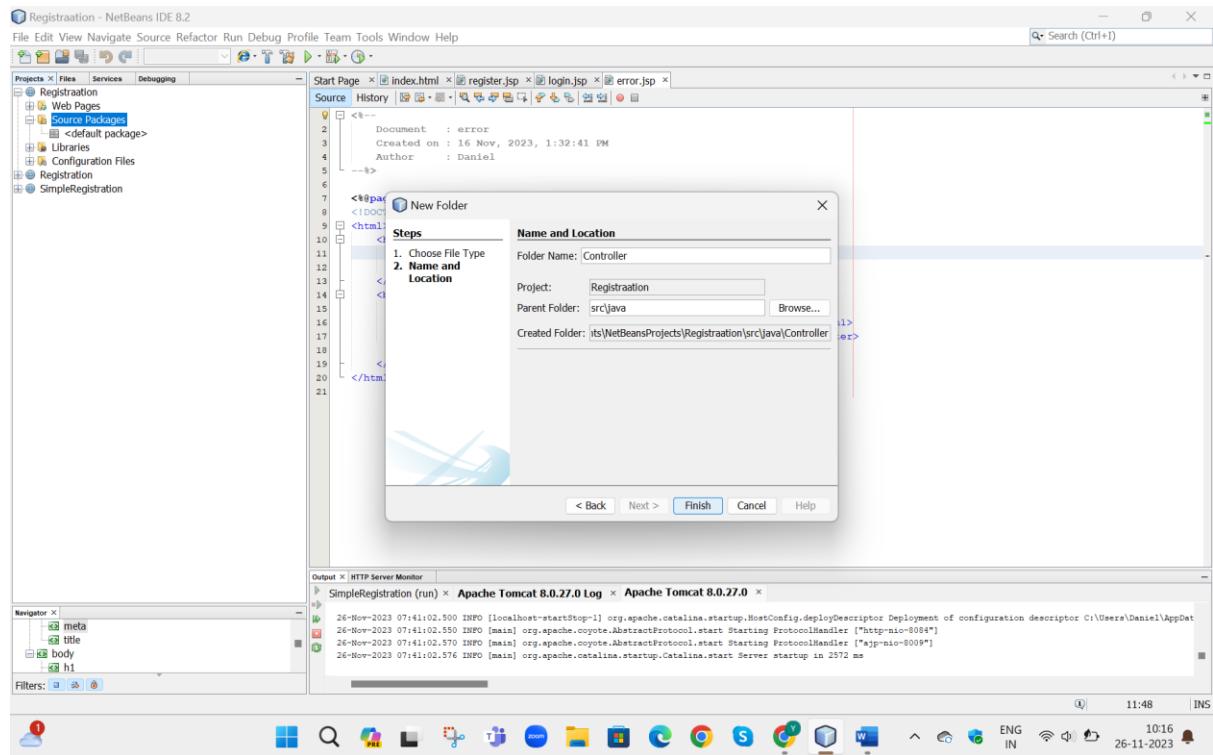
Hit **Finish** button and enter the code as shown below:



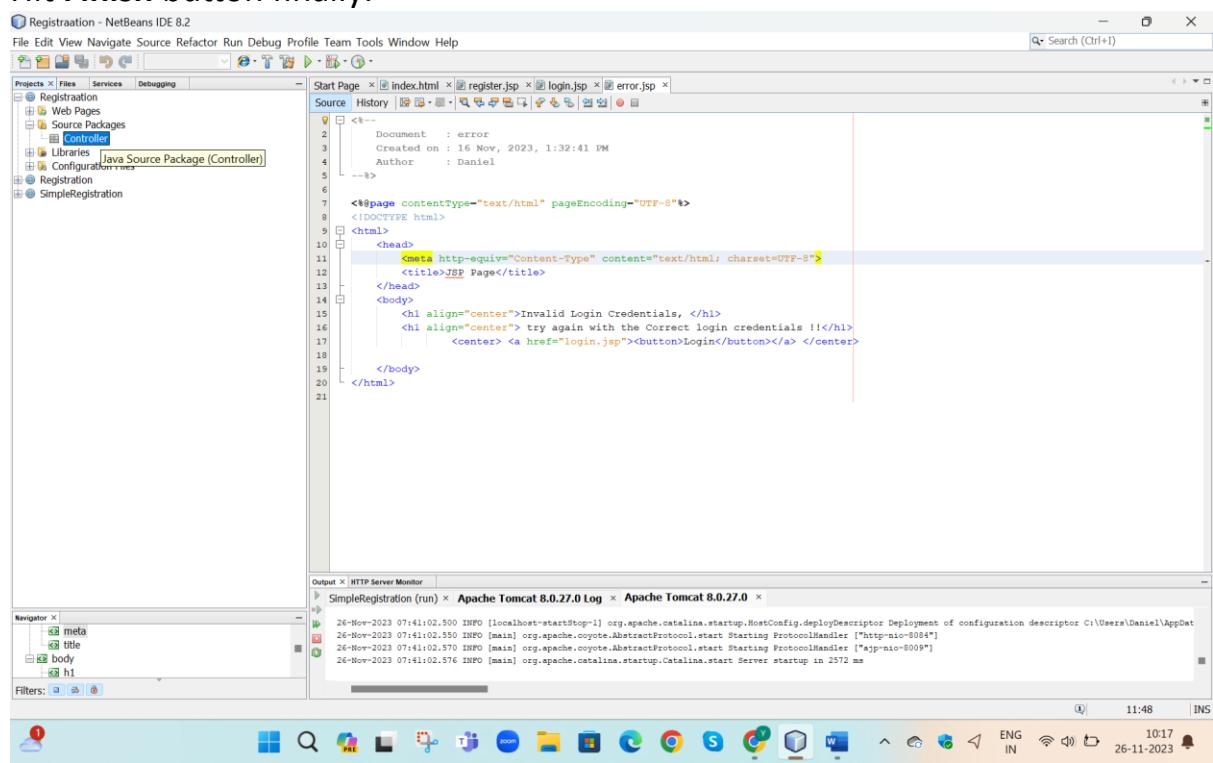
Creating .java files(Servlets)



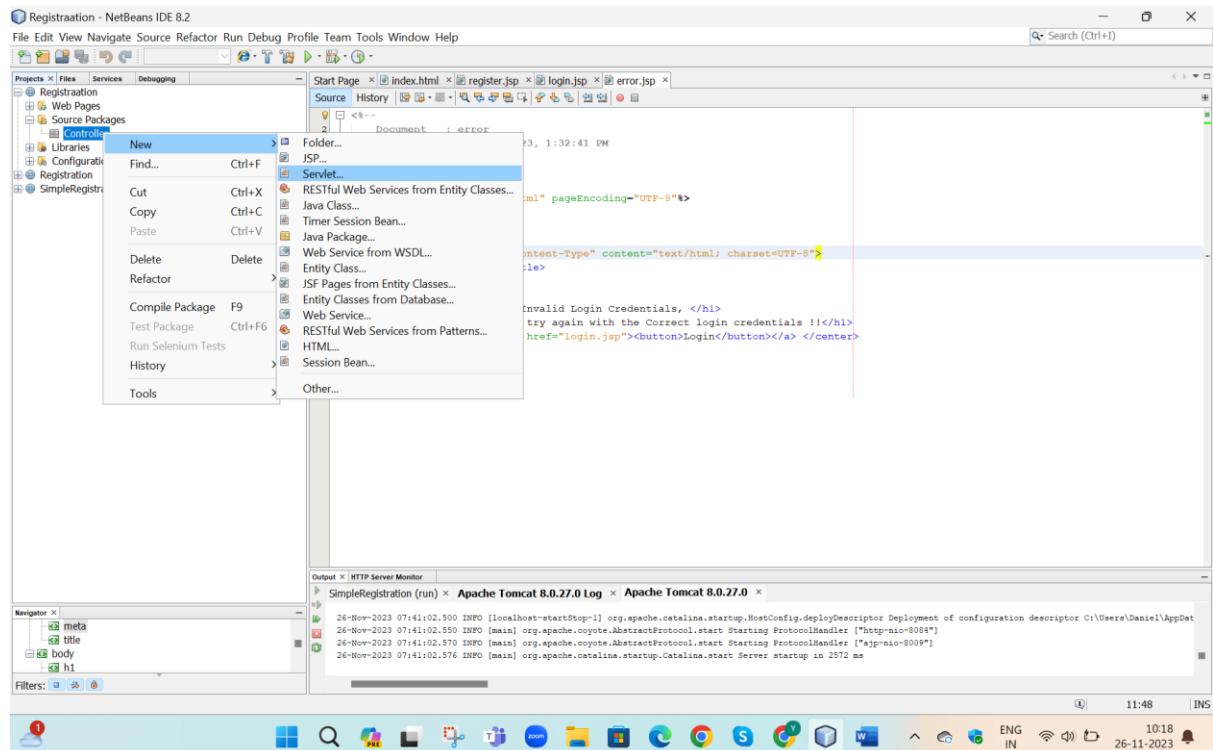
Before creating Servlets, make sure to create **Folder name** with “**Controller**” as displayed below:



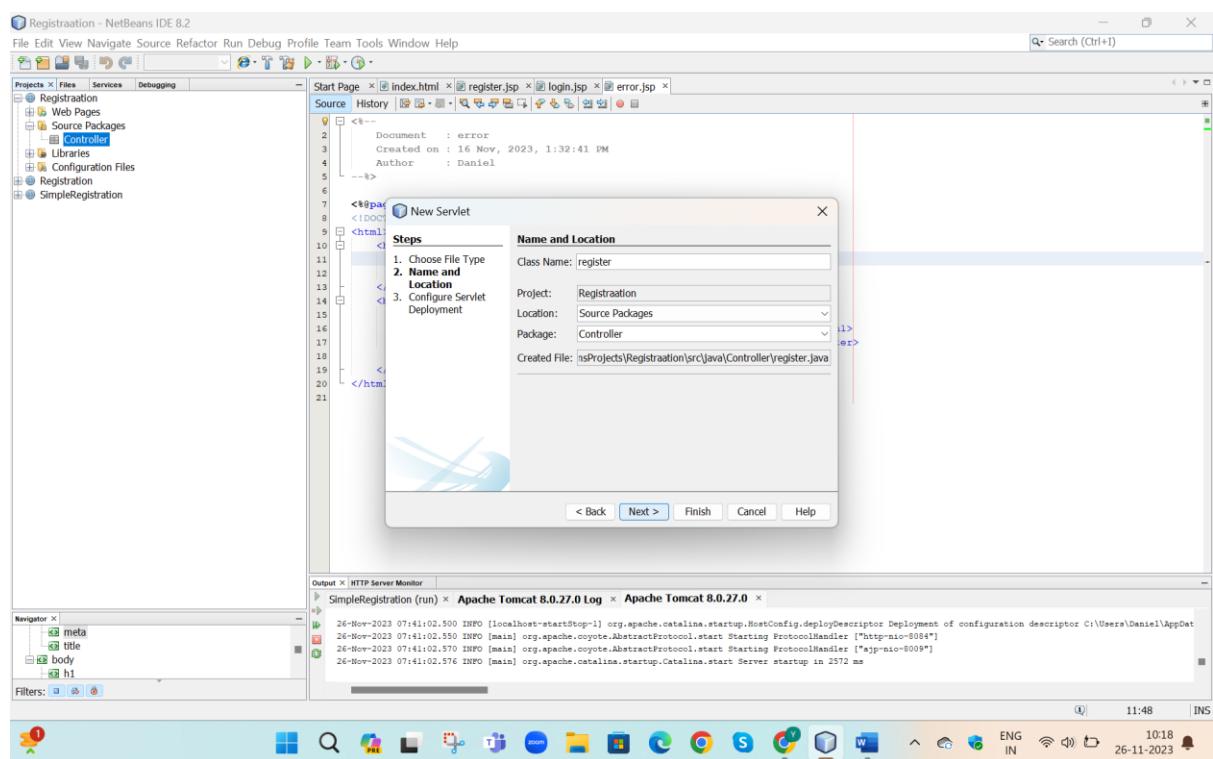
Hit Finish button finally.

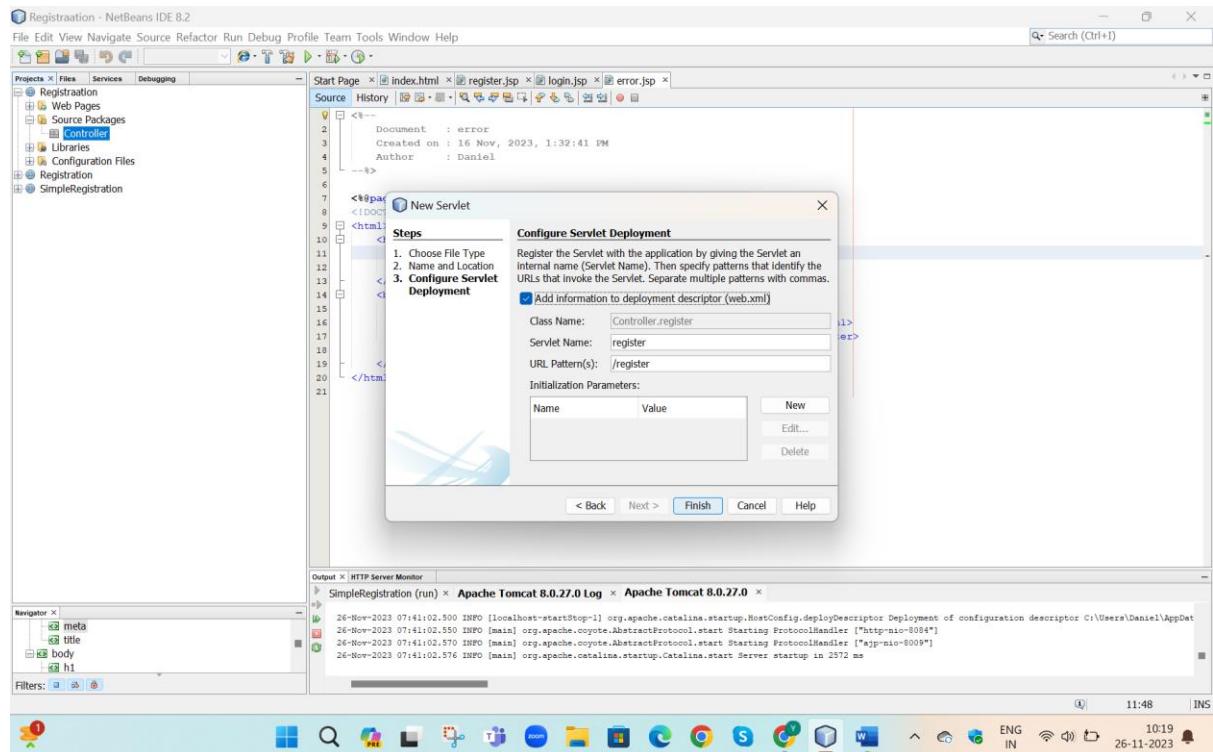


Now to create Servlets page:

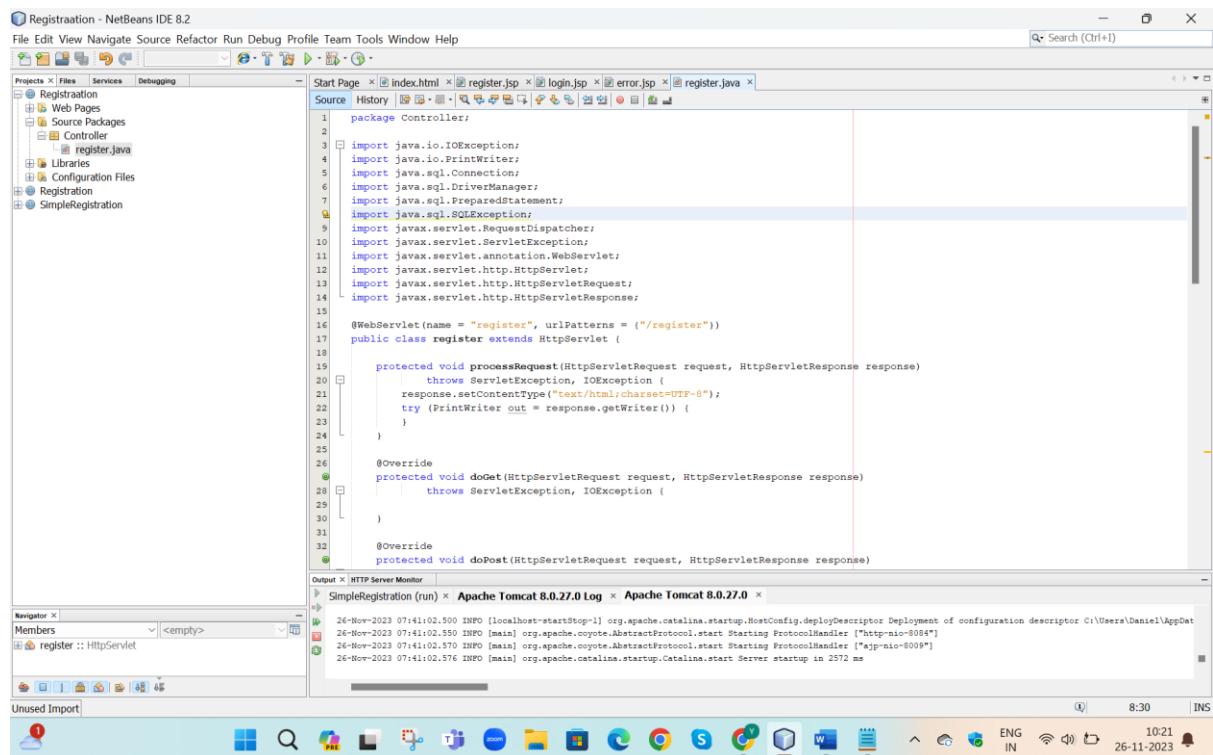


File name with “register” as shown below:

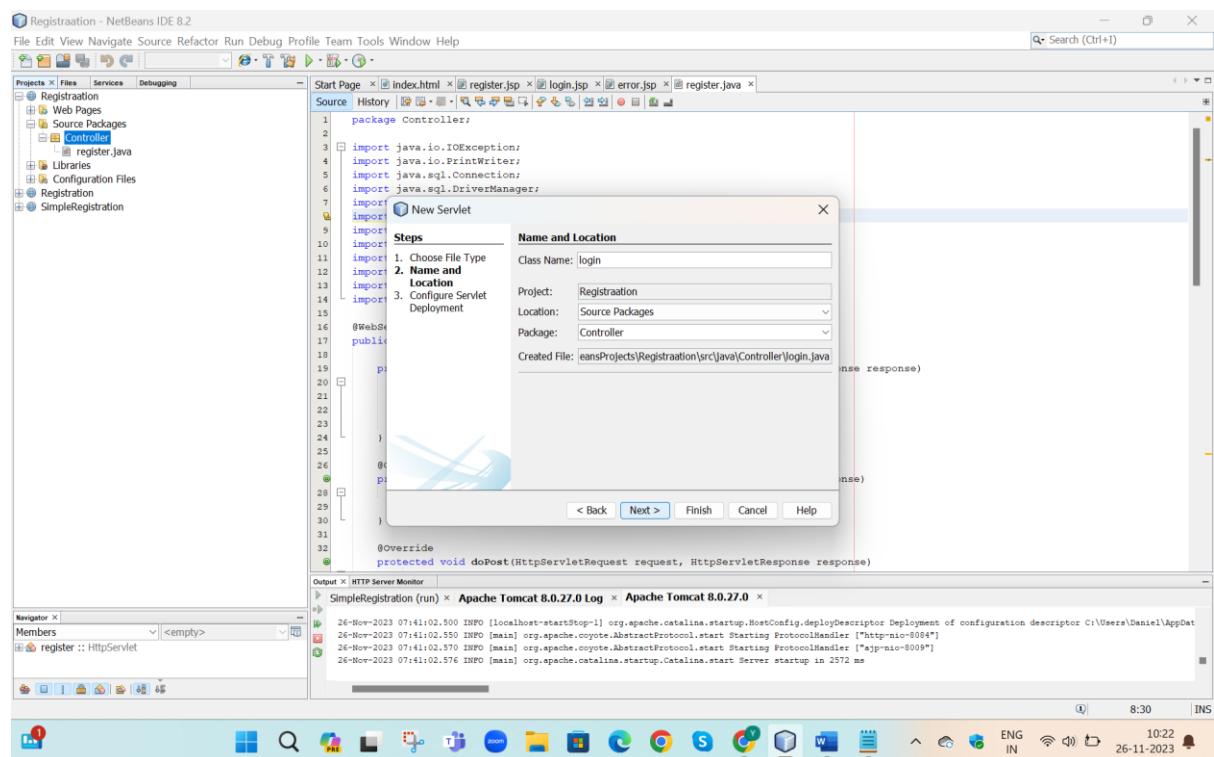
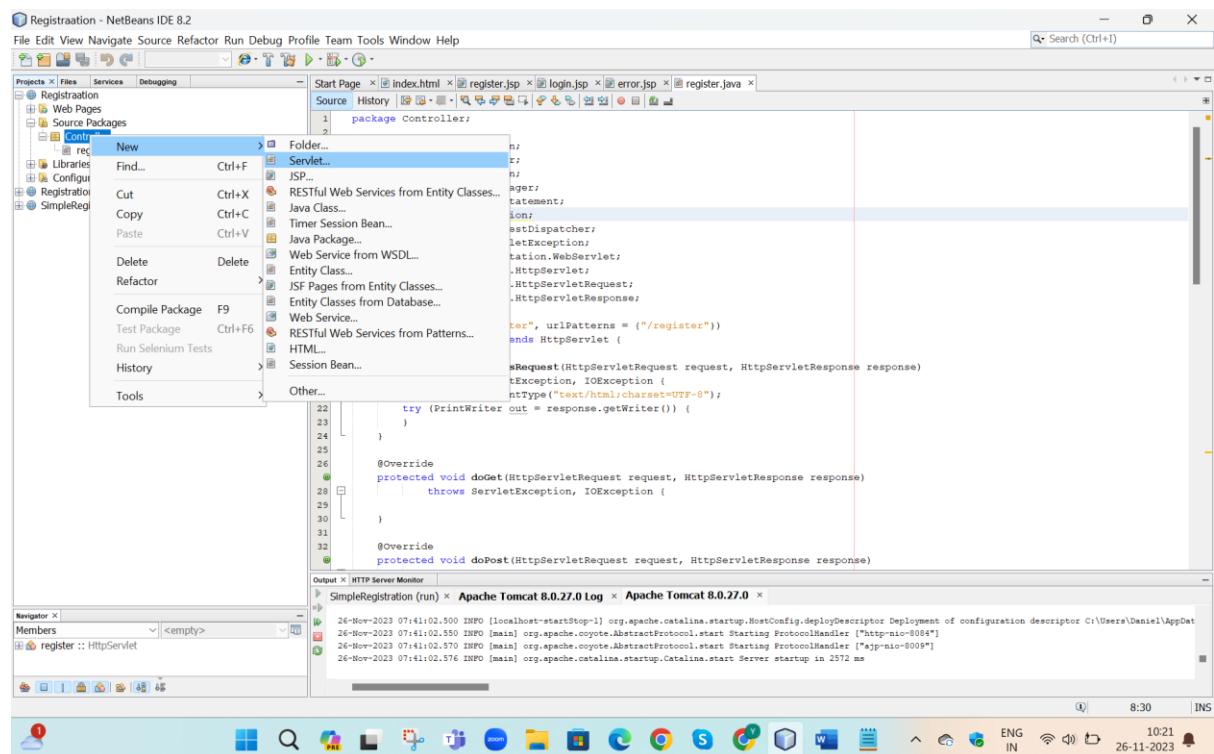




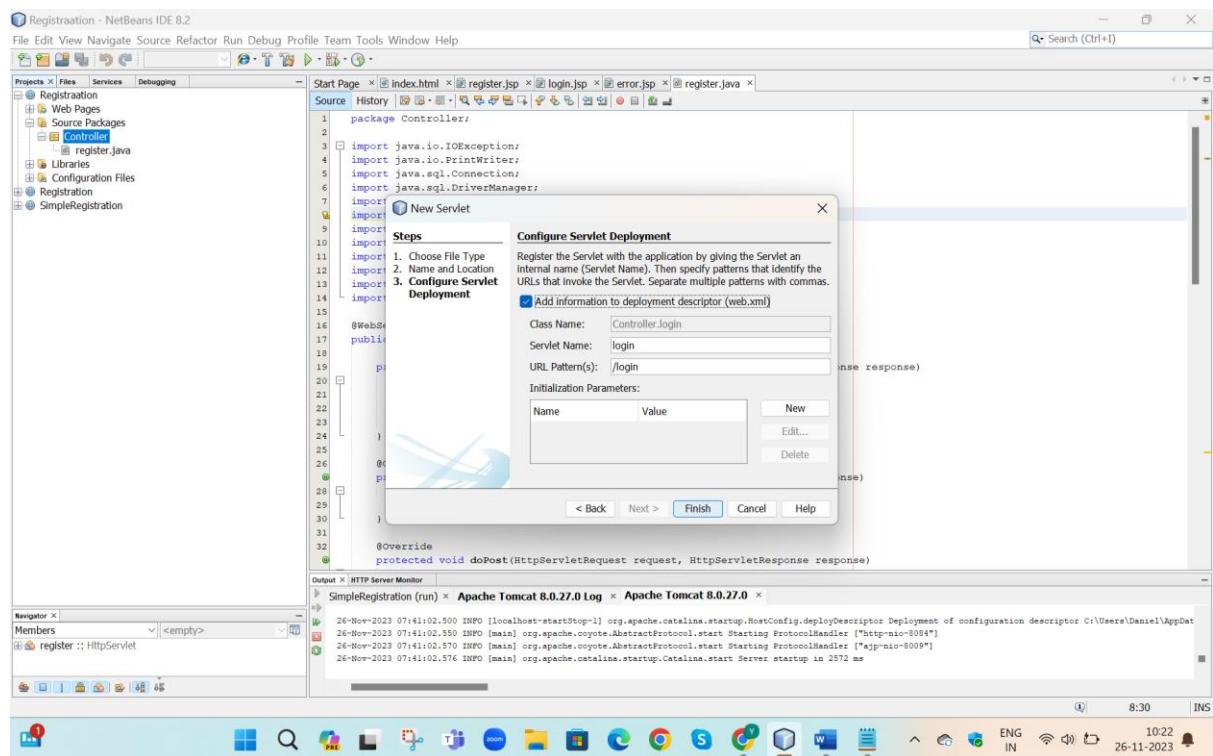
Add checkbox and Hit **Finish** and enter the code by removing the default code from it as shown below:



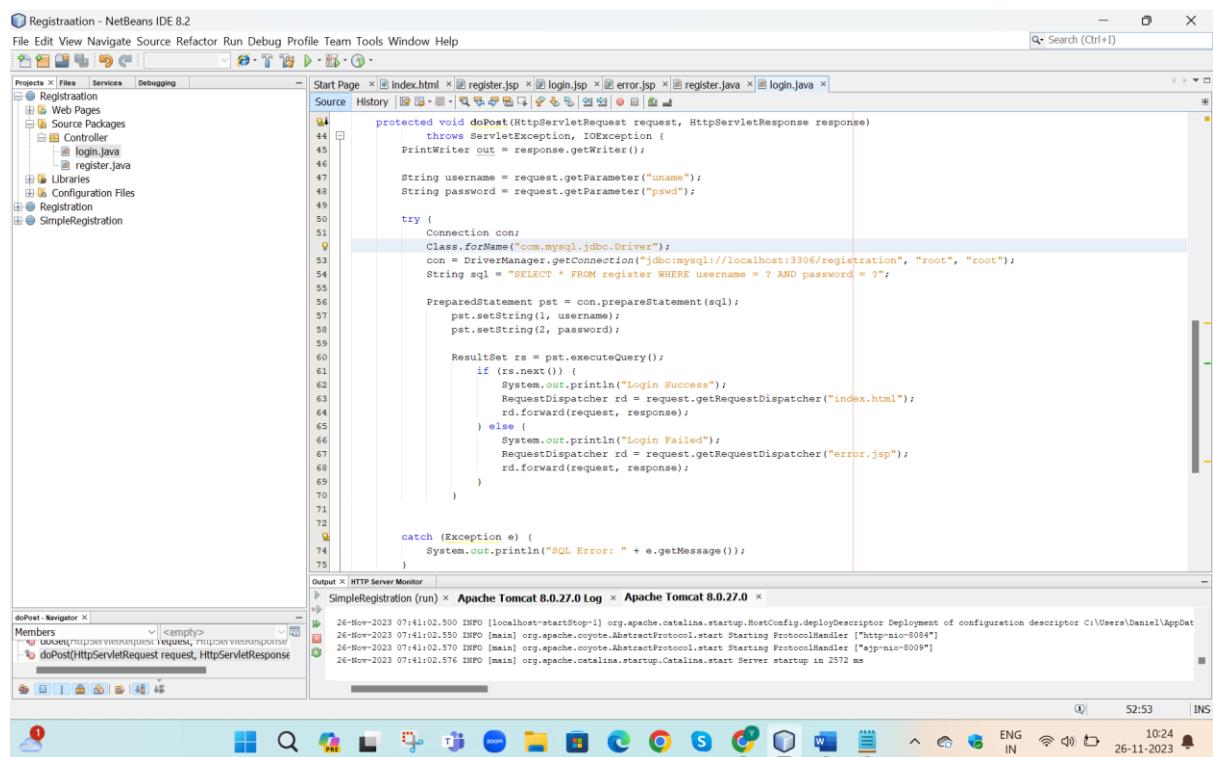
Again, create another file name with “login” as shown below:



Hit **next** and



Mark the checkbox “Add Information(web.xml)” and hit **Finish** button and enter the code for it by removing the default code as shown below:



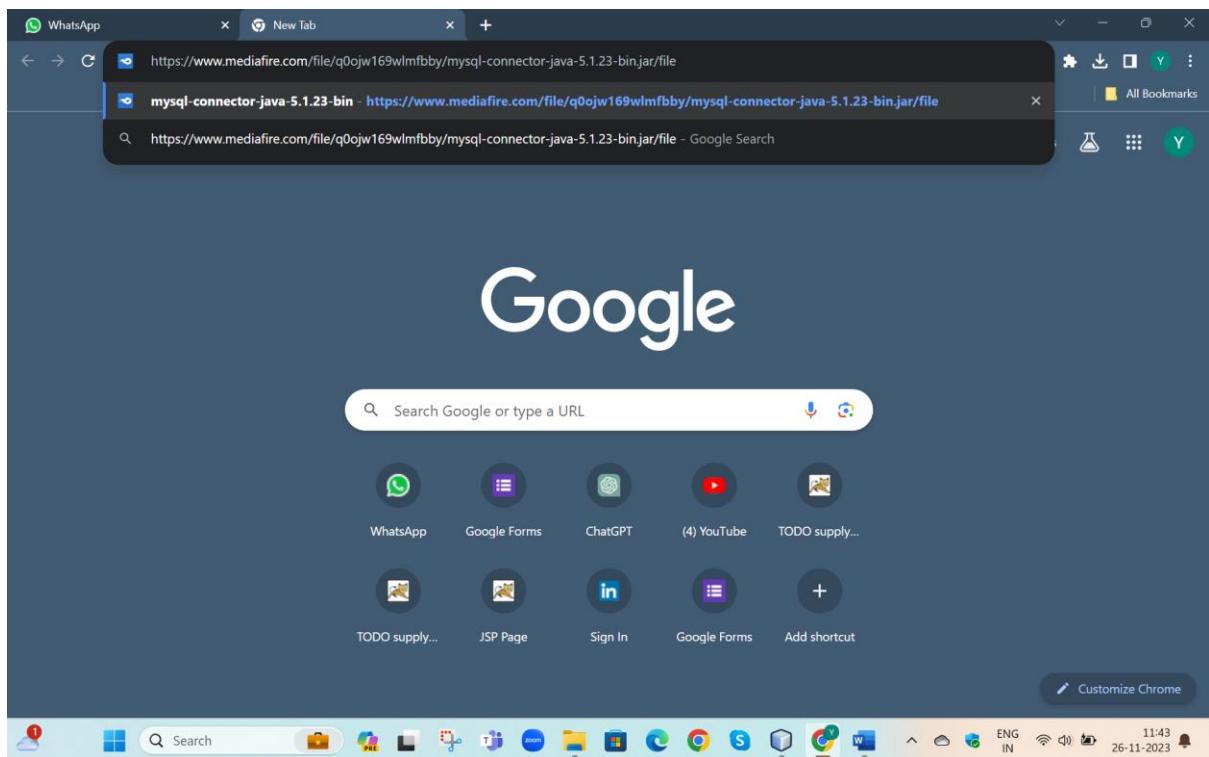
How to add Jars (Supported files):

----- > Download Jar files on your System using the below links:

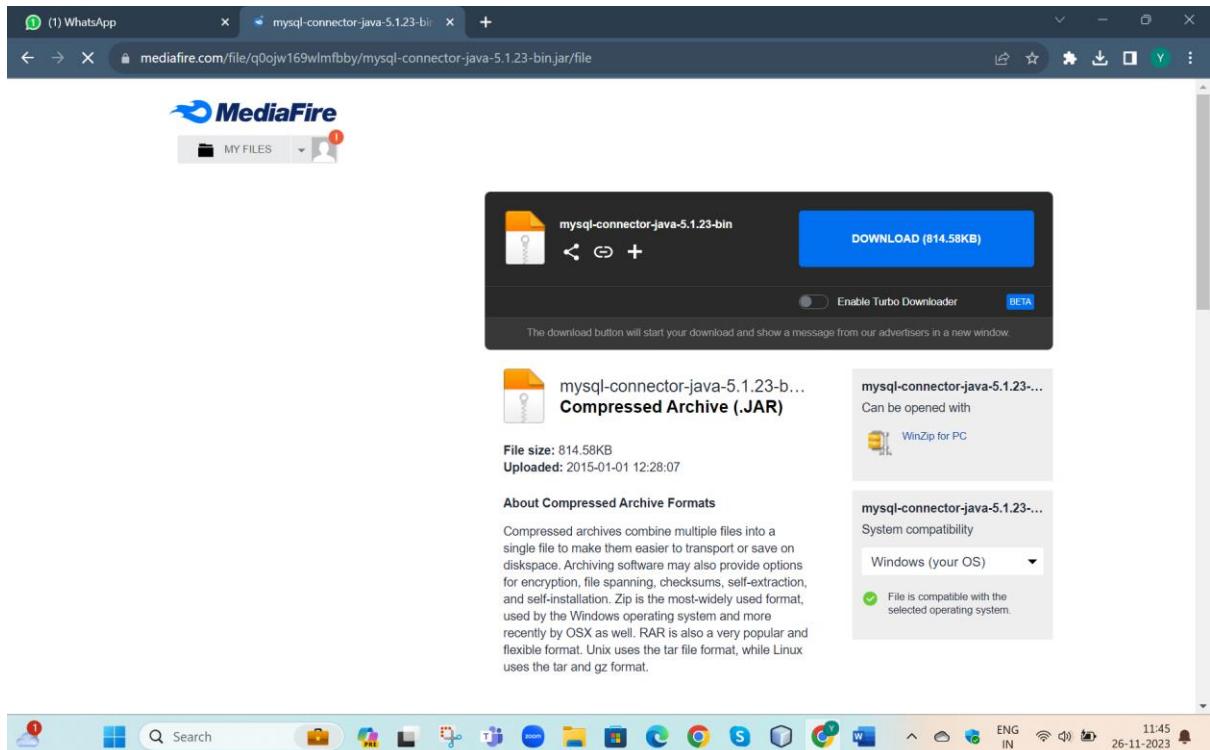
Links to Download Jars:

1. <https://www.mediafire.com/file/q0ojw169wlmbby/mysql-connector-java-5.1.23-bin.jar/file>
2. <https://www.mediafire.com/file/f3l49kv9xczcpco/cos-multipart.jar/file>
3. <https://www.mediafire.com/file/kynp64sxy40vlvh/gson-2.2.2.jar/file>
4. <https://www.mediafire.com/file/rah5o9dh7yqces6/java-mail-1.4.4.jar/file>

Copy each link and paste in browser as shown below:



This will redirect you to media fire website, there you will find Download button, just hit **Download** button to download the respective jar files as shown below:



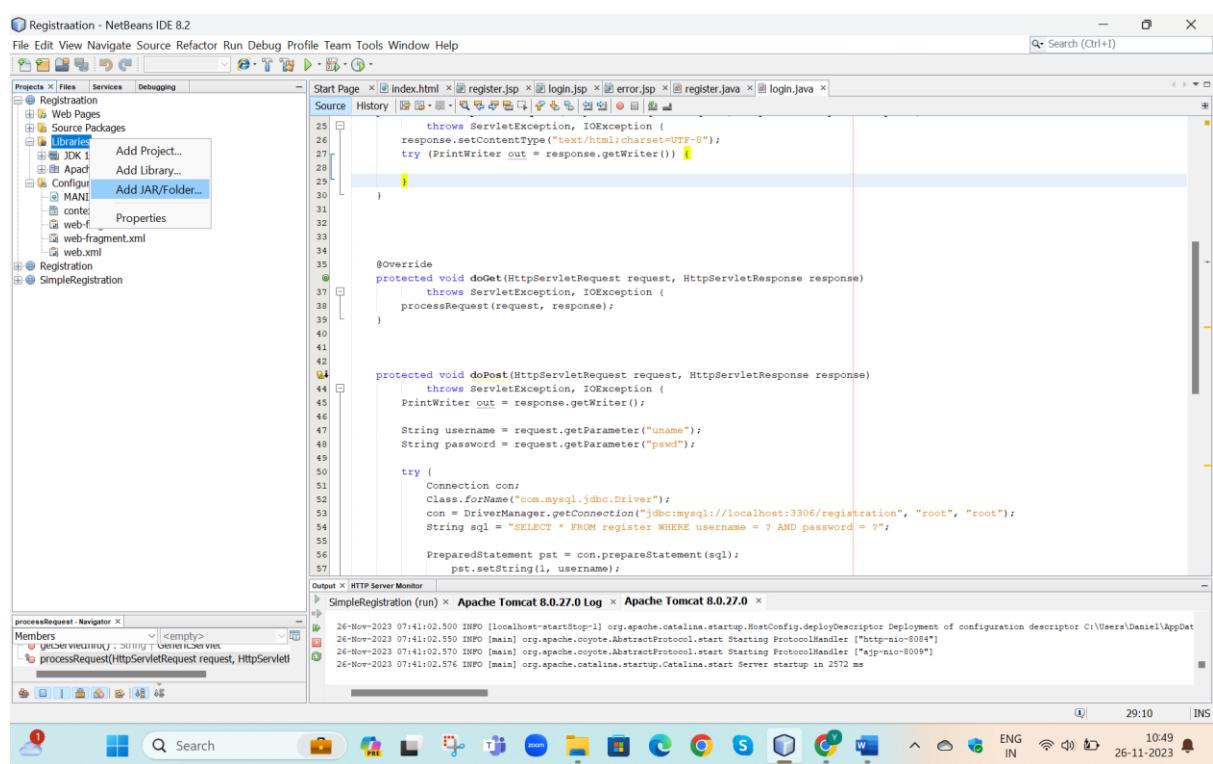
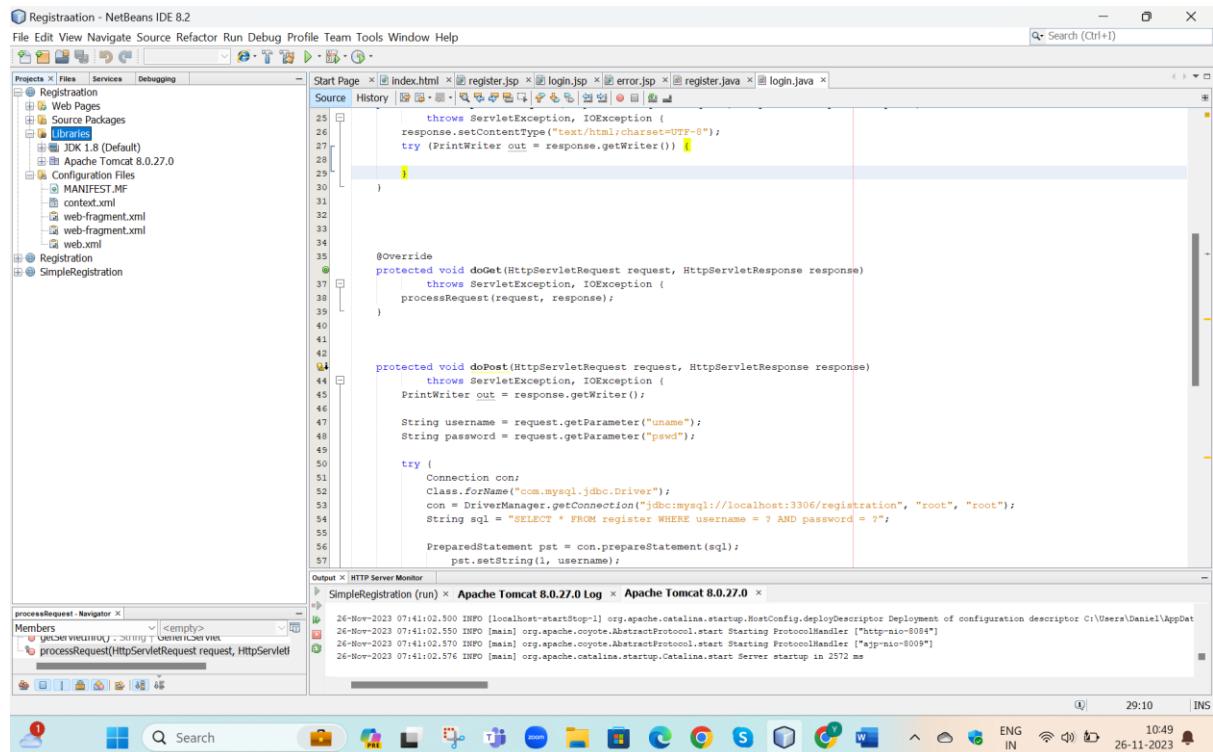
The respective files will be downloaded for you as soon as you click on **Download** button.

Hence, **Repeat the same process for downloading the other 3 jar files.**

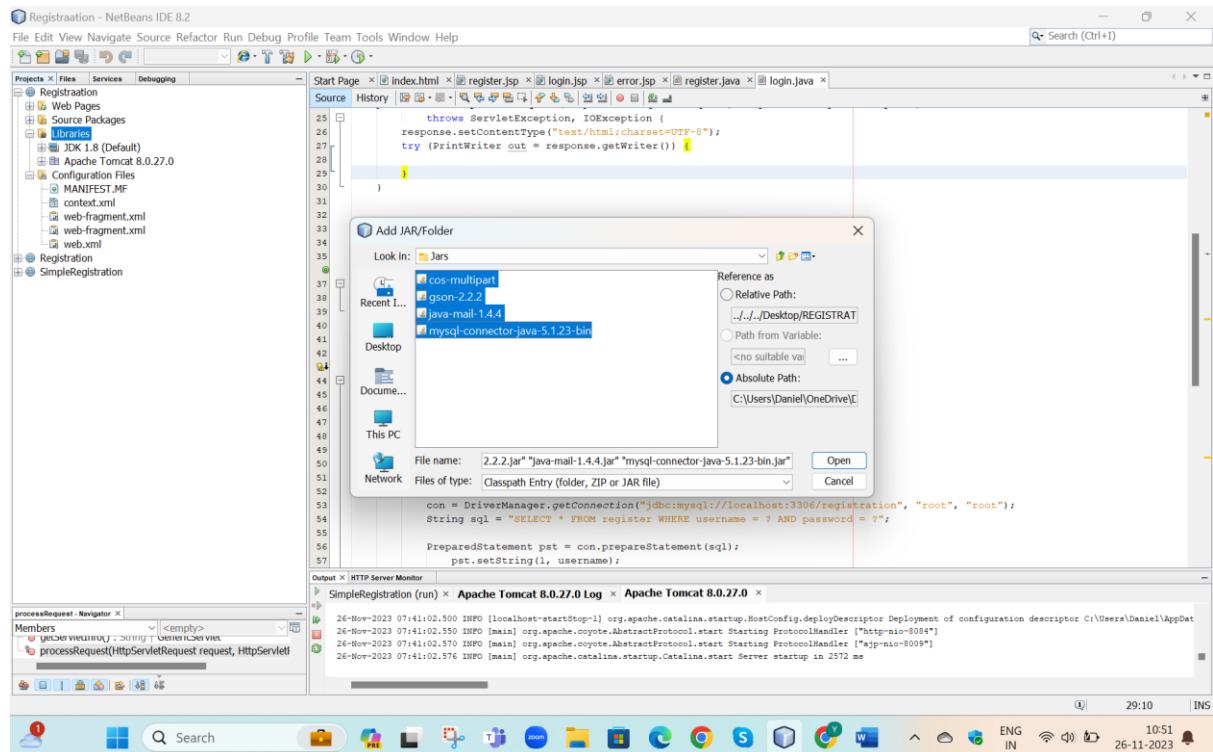
After downloaded these 4 files, you have to upload these files to our NetBeans Project that we have created so far as shown below:

For **Uploading jars**, follow the below process:

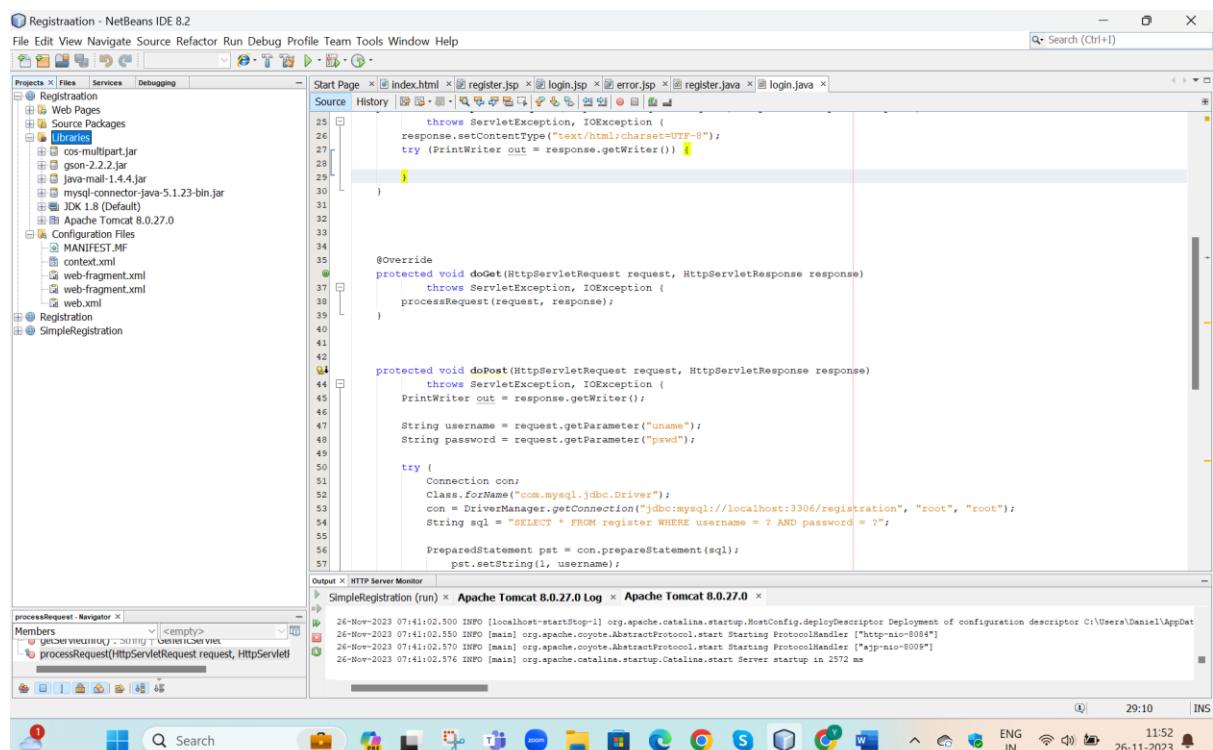
Select **Libraries** Folder -> Right Click -> Choose option (**Add JAR/Folder**) as shown below:



After clicking on **Add JAR/Folder** option, the following dialogue box will appear, you have to navigate to the location where you had downloaded the jar files, which enables us to upload jar files as displayed in the screenshot below:



Select all 4 necessary downloaded files and click on **Open** button to upload. Hence, you will see jar files being added to Libraries Folder, as soon as you click on **Open** button as displayed below:



In this way, you can add Jar Files to your NetBeans Project.

Source Code:

HTML Files:

index.html:

```
<html>
<head>
    <title>Home </title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        .container{
            width: 100%;
            height: 50px;
            background: rgba(19, 160, 170, 0.822);
        }
        li
        {
            float: left;
            text-decoration: none;
            list-style: none;
            padding: 10px 10px 10px;
        }
        li a{
            color: rgb(255, 255, 255);
            text-decoration: none;
        }
        li a:hover{
            background-color : rgba(0, 242, 255, 0.719);
            transition: background-color 3 s;
        }
        body {
            background-color:rgba(35, 200, 230, 0.999) ;
            margin: 2;
            display: flex;
        }
        nav a {
            border-bottom: 2px solid transparent;
            transition: border-bottom 0.3s;
        }
    </style>
</head>
<body>
    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </nav>
</body>
</html>
```

```
nav a:hover {  
    border-bottom: 5px solid rgb(226, 42, 82);  
}  
    </style>  
</head>  
<body>  
    <div class = "container">  
        <nav>  
            <ul>  
                <li><a href = "index.html" >HOME</a></li>  
                <li><a href = "index.html" >ABOUT</a></li>  
                <li><a href = "index.html" >CONTACT</a></li>  
                <li style="float: right;"><a href = "register.jsp" >REGISTER</a></li>  
                <li style="float: right;"><a href = "login.jsp" >LOGIN</a></li>  
                <li><a href = "index.html" >OUR SERVICES</a></li>  
            </ul>  
        </nav>  
    </div>  
</body>  
</html>
```

JSP Files:

register.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
        <title>JSP Registration Page</title>  
        <style>  
            .registerconatiner{  
                width: 100%;  
                height: 900%;  
                display: flex;  
                justify-content: center;  
                align-items: center;  
            }  
            table {  
                width: 120%;  
                height: 60%;  
                background-color: rgba(37, 37, 222, 0.822);  
                box-shadow: 0 0 20px rgba(255, 98, 0, 0.974);  
                border-radius: 20px;  
            }
```

```
color: rgb(255, 255, 255);
}
input[type="submit"] {
background-color: rgb(5, 177, 235);
}
input[type="reset"] {
background-color: rgb(12,182,232);
}
body {
background-color: rgba(29, 66, 203, 0.723);
}
</style>
</head>
<body>
<div class = "registerconatiner">
<form method="post" action="register">
<center>
<table>
<tr>
<th colspan="2">Registration Form</th>
</tr>
<tr>
<td>Full Name:</td>
<td><input type="text" placeholder="Enter Your Name" name="fname" required=""></td>
</tr>
<tr>
<td>User Name:</td>
<td><input type="text" placeholder="Enter User Name" name="uname" required=""></td>
</tr>
<tr>
<td>Email:</td>
<td><input type="email" placeholder="Enter Your Email" name="email" required=""></td>
</tr>
<tr>
<td>Password:</td>
<td><input type="password" placeholder="Enter Your Password" name="pswd" required=""></td>
</tr>
```

```

<tr>
    <td>Confirm Password:</td>
    <td><input type="password" placeholder="Reconfirm Password"
name="cfmpswd" required=""></td>
</tr>
<tr>
    <th colspan="1"><input type="submit" value="Register"></th>
    <th colspan="1"><input type="reset" value ="reset"> </th>
</tr>
</table>
</center>
</form>
</div>
</body>
</html>

```

login.jsp:

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Login Page</title>
        <style>
            table{
                background-color: rgba(37, 37, 204, 0.822);
                box-shadow: 0 0 10px rgba(255, 98, 0, 0.974);
                border-radius: 20px;
                height:100%;
                width:30%;
                color: antiquewhite;
                text-align: center;
            }
            input[type="submit"] {
                background-color: rgb(5, 177, 235);
            }
            input[type="reset"] {
                background-color: rgb(15, 182, 232);
            }
            body {
                background-color: rgba(29, 66, 203, 0.723);
            }
        </style>
    </head>
    <body>
        <table border="1">
            <tr>
                <td>User Name:</td>
                <td><input type="text" name="username" value="Enter User Name" /></td>
            </tr>
            <tr>
                <td>User Password:</td>
                <td><input type="password" name="password" value="Enter Password" /></td>
            </tr>
            <tr>
                <td>Confirm Password:</td>
                <td><input type="password" placeholder="Reconfirm Password"
name="cfmpswd" required=""></td>
            </tr>
            <tr>
                <th colspan="2"><input type="submit" value="Login"></th>
                <th colspan="2"><input type="reset" value ="Reset"> </th>
            </tr>
        </table>
    </body>
</html>

```

```
</style>
</head>
<body>
    <form method="post" action="login">
        <center>
            <table>
                <tr>
                    <th colspan="2" ><u>LOGIN</u></th>
                </tr>
                <tr>
                    <td> Username </td>
                    <td><input type="text" placeholder="ENTER USER NAME" name="uname" required=""></td>
                </tr>
                <tr>
                    <td>Password</td>
                    <td><input type="password" placeholder="ENTER YOUR PASSWORD" name="pswd" required=""></td>
                </tr>
                <tr>
                    <th colspan="1"><input type="submit" value="Login"> </th>
                    <th colspan="1"><input type="reset" value= "reset"></th>
                </tr>
            </table>
        </center>
    </form>
</body>
</html>
error.jsp:
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Login error Page</title>
        <style>
            body{
                background-color: rgba(234,34,75,0.888);
            }
        </style>
    </head>
```

```

<body>
    <h1 align="center"><b> Invalid Login Credentials, </b> </h1>
    <h1 align="center"><b> try again with the Correct login credentials !!</b>
</h1>
    <center> <a href="login.jsp"><button>Login</button></a> </center>
</body>
</html>

```

SERVLETS (. Java Files):

register.java:

```

package Controller;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "register", urlPatterns = {" /register"})
public class register extends HttpServlet
{
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }
    @Override

```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
String fullname = request.getParameter("fname");
String username = request.getParameter("uname");
String email = request.getParameter("email");
String password = request.getParameter("pswd");
String confirmpassword = request.getParameter("cfmpswd");

try {
    Connection con;
    Class.forName("com.mysql.jdbc.Driver");
    con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/registration",
"root", "root");
    System.out.println("Connected to Database");
    String sql = "INSERT INTO register (fullname, username, email, password,
confirmpassword) VALUES (?, ?, ?, ?, ?)";
    PreparedStatement pst = con.prepareStatement(sql);
    pst.setString(1, fullname);
    pst.setString(2, username);
    pst.setString(3, email);
    pst.setString(4, password);
    pst.setString(5, confirmpassword);
    pst.executeUpdate();
    System.out.println("Registration Success");
    RequestDispatcher rd = request.getRequestDispatcher("index.html");
    rd.forward(request, response);
}
catch (Exception e)
{
    System.out.println("Error :: " + e.getMessage());
}
}

@Override
public String getServletInfo()
{
    return "Short description";
}
```

login.java:

```
package Controller;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.PreparedStatement;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "login_1", urlPatterns = {"/login_1"})
public class login extends HttpServlet
{
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            }

    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        String username = request.getParameter("uname");
        String password = request.getParameter("pswd");

        try {
            Connection con;
```

```
Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/registration",
"root", "root");
String sql = "SELECT * FROM register WHERE username = ? AND
password = ?";
PreparedStatement pst = con.prepareStatement(sql);
pst.setString(1, username);
pst.setString(2, password);
ResultSet rs = pst.executeQuery();
if (rs.next())
{
    System.out.println("Login Success");
    RequestDispatcher rd =
request.getRequestDispatcher("index.html");
    rd.forward(request, response);
}
else
{
    System.out.println("Login Failed");
    RequestDispatcher rd =
request.getRequestDispatcher("error.jsp");
    rd.forward(request, response);
}
}
catch (Exception e)
{
    System.out.println("SQL Error: " + e.getMessage());
}
}
@Override
public String getServletInfo()
{
    return "Login Servlet";
}
}
```

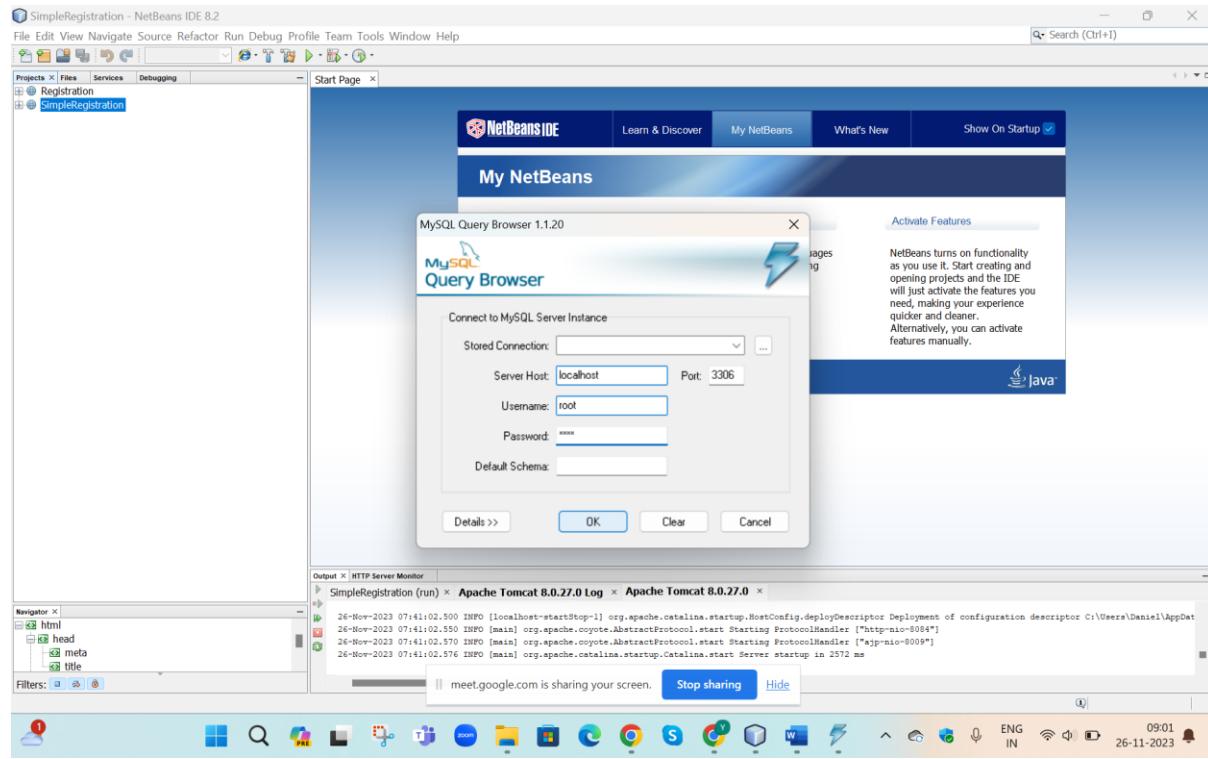
Database Querying:

Open MySQL Query Browser and enter the following login credentials to logon to MySQL Query Browser:

Server Host: localhost

Username: root

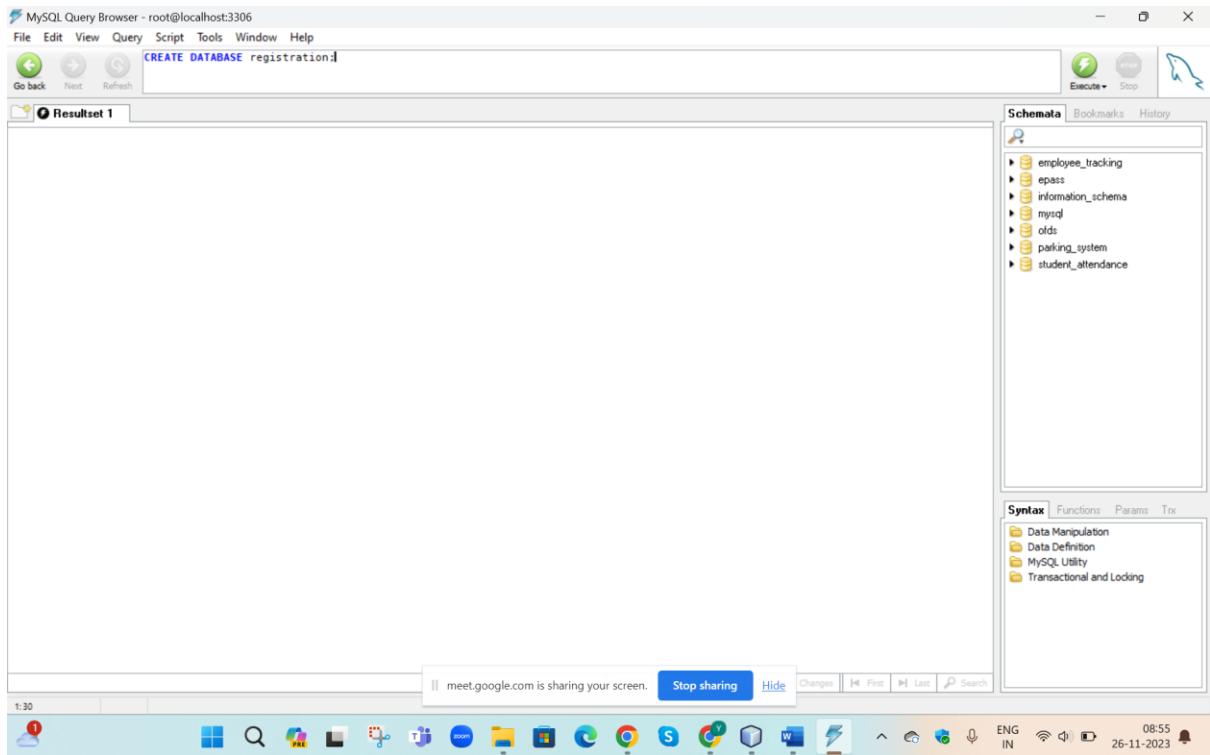
Password: root



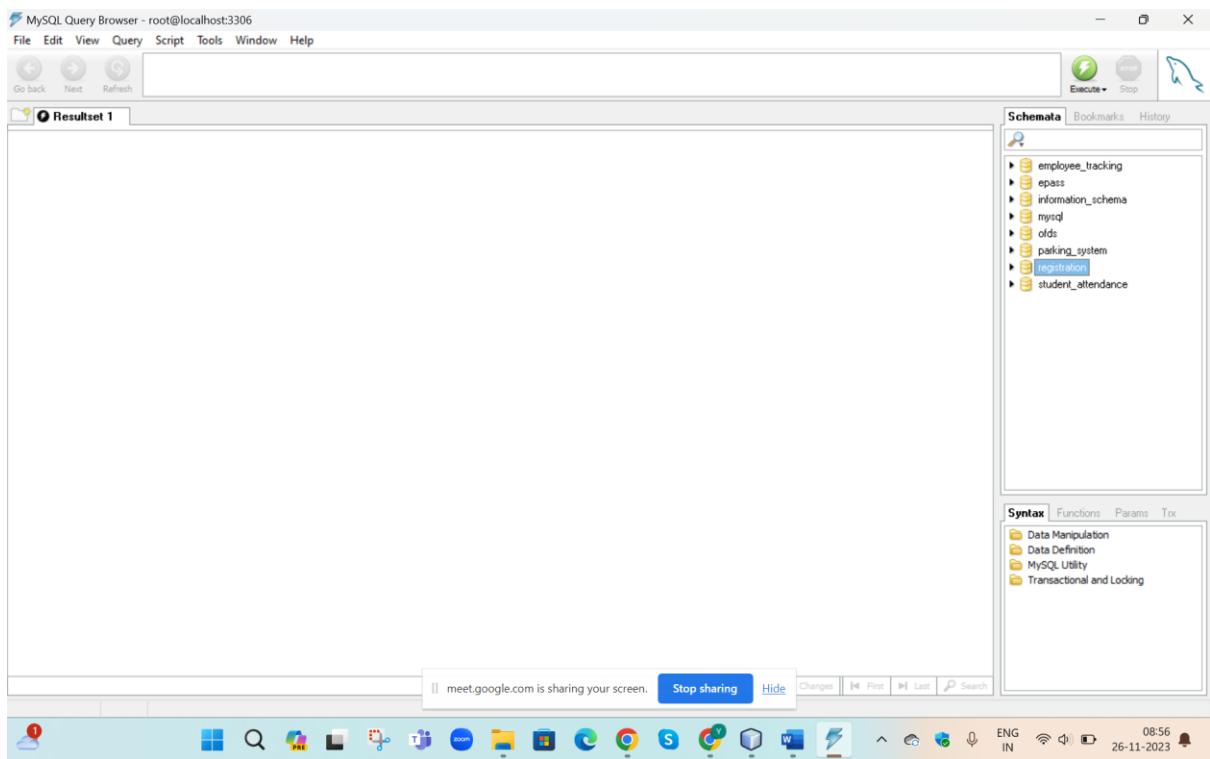
MySQL commands as follows:

1. To create Database, use the following command and hit execute button which is visible on the right-hand side corner:

CREATE DATABASE registration;

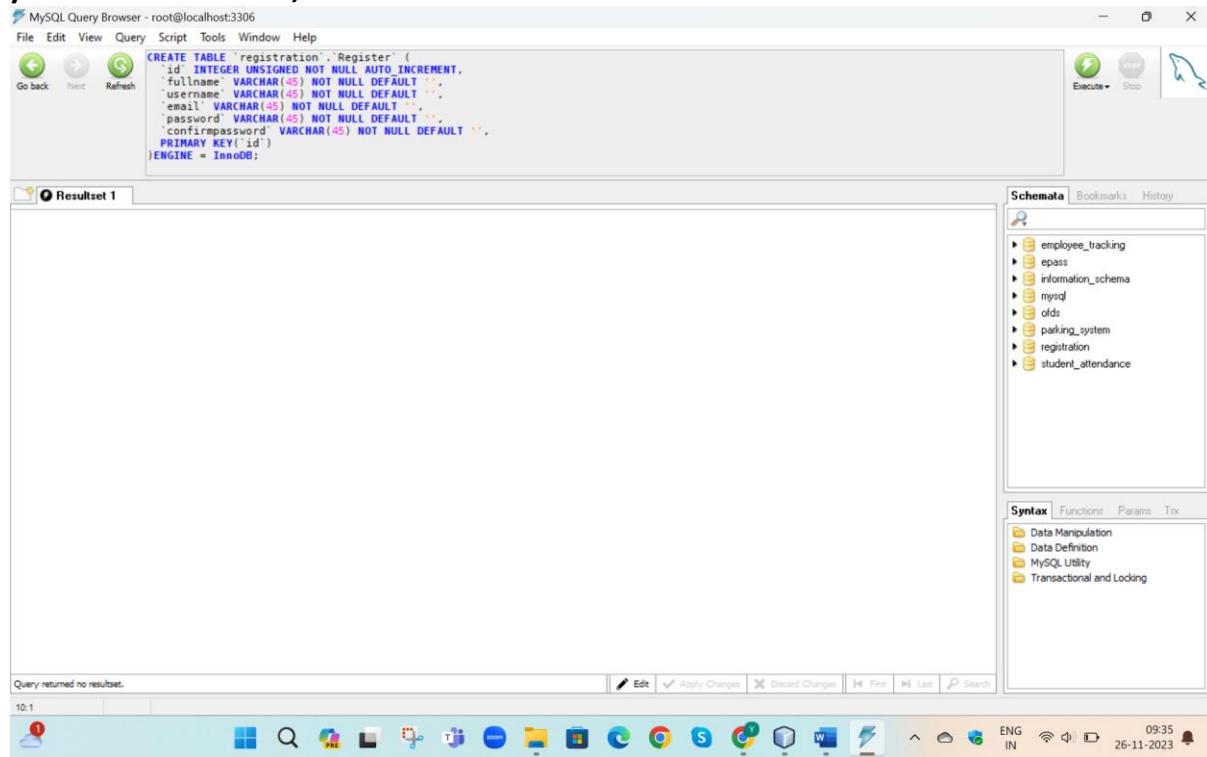


After successfully executing this query, the SQL query will create Database name with “**registration**” as shown below (if changes not reflected, just close this query browser and open again)

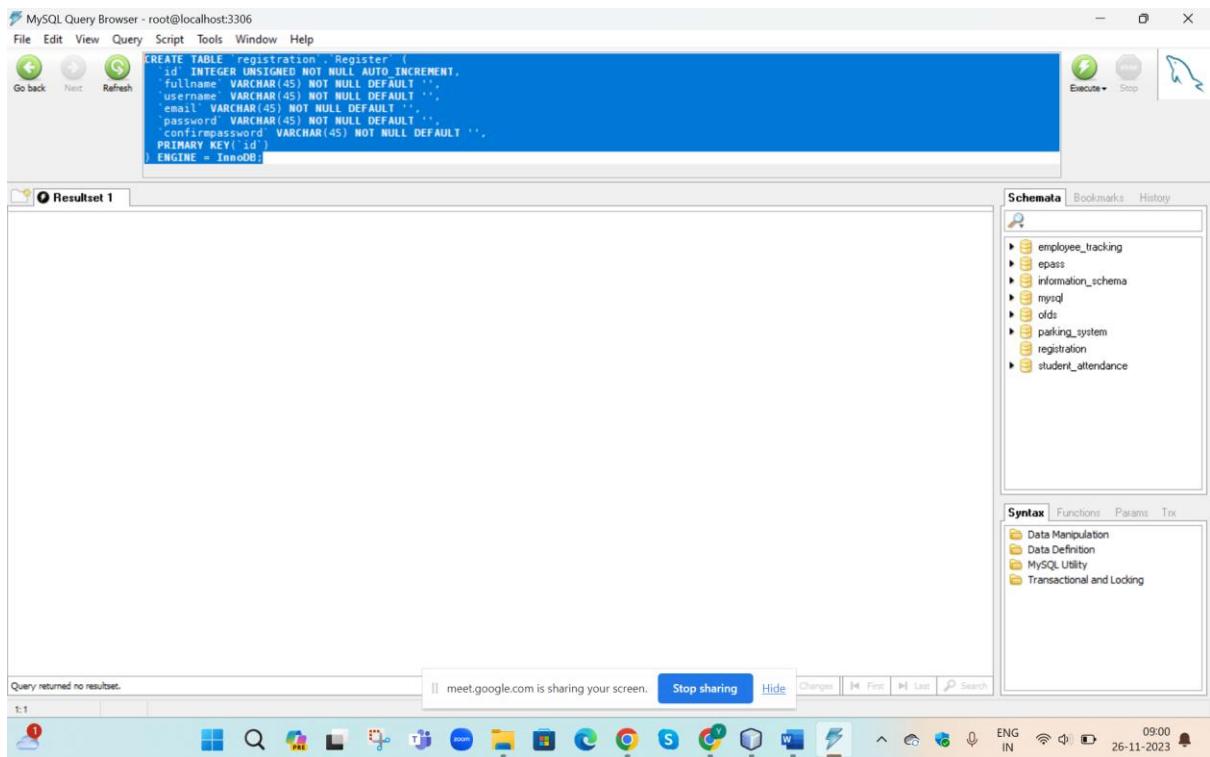


2. To create “register” table in “registration” Database, use the following query:

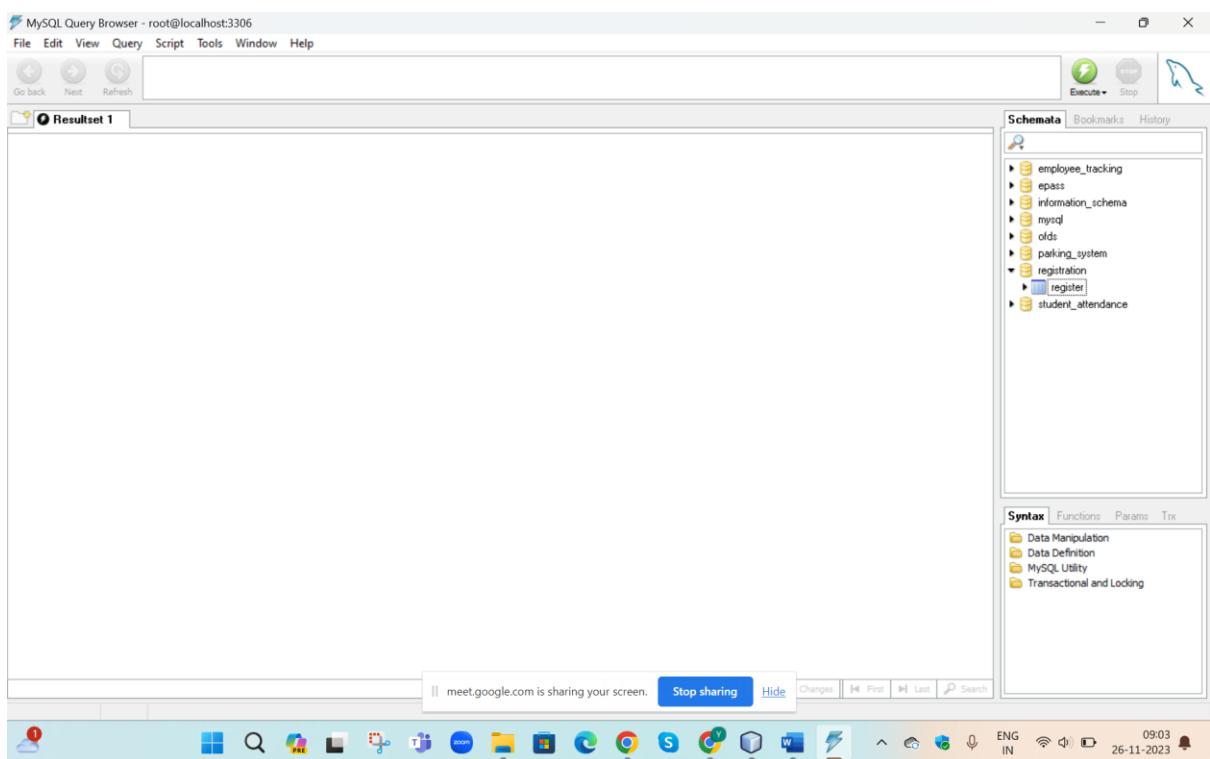
```
CREATE TABLE `registration`.`Register` (
`id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
`fullname` VARCHAR(45) NOT NULL DEFAULT '',
`username` VARCHAR(45) NOT NULL DEFAULT '',
`email` VARCHAR(45) NOT NULL DEFAULT '',
`password` VARCHAR(45) NOT NULL DEFAULT '',
`confirm password` VARCHAR(45) NOT NULL DEFAULT '',
PRIMARY KEY(`id`)
)ENGINE = InnoDB;
```



enter the query as specified above and hit execute button.



Upon successfully executing this query, we will see Database table created with name “**register**” as shown below (if changes not reflected, just close this query browser and open again)



3. To retrieve the data from the “register” table, use the below command:

```
SELECT * FROM `registration`.`register`;
```

The screenshot shows the MySQL Query Browser interface. The query window contains the command: `SELECT * FROM `registration`.`register`;`. The results pane, titled "Resultset 1", is empty, indicating 0 rows fetched. The schema browser on the right shows the database structure, including the `registration` schema which contains the `register` table. The system tray at the bottom shows various icons and the date/time: 26-11-2023, 09:04.

4. To delete the rows(data) from the table “register”, use the following command:

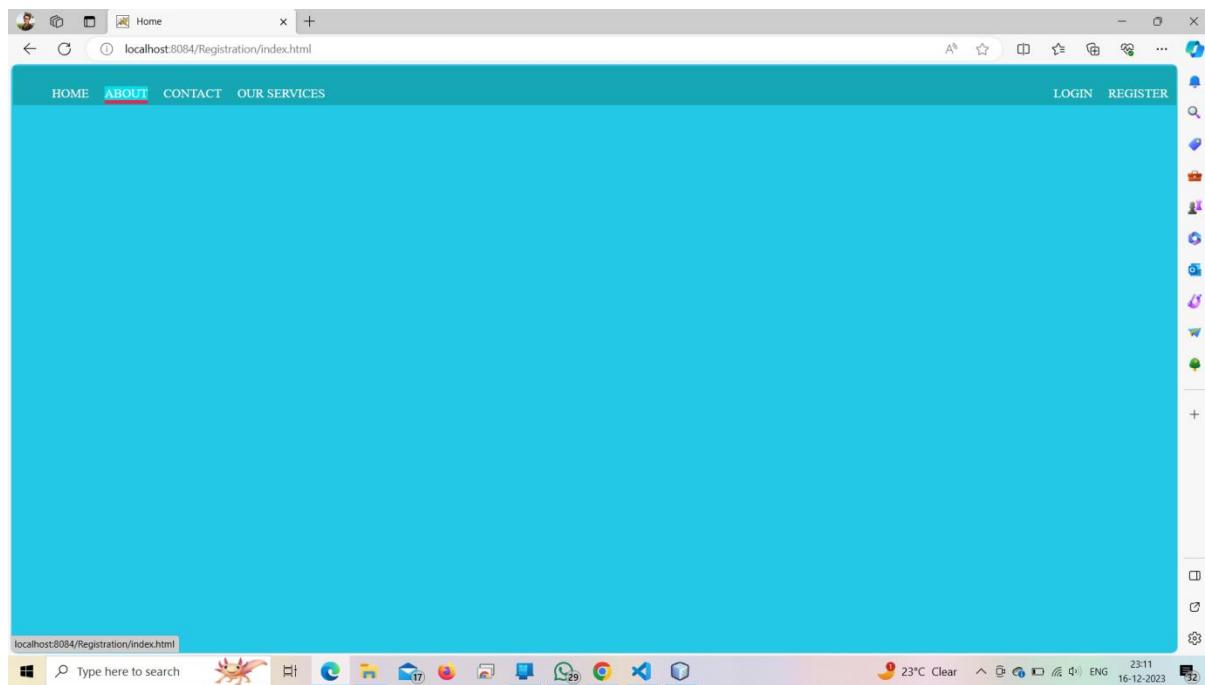
```
delete FROM `registration`.`register`;
```

5. To start a new record with serial number “1” after performing delete operation, use the following command:

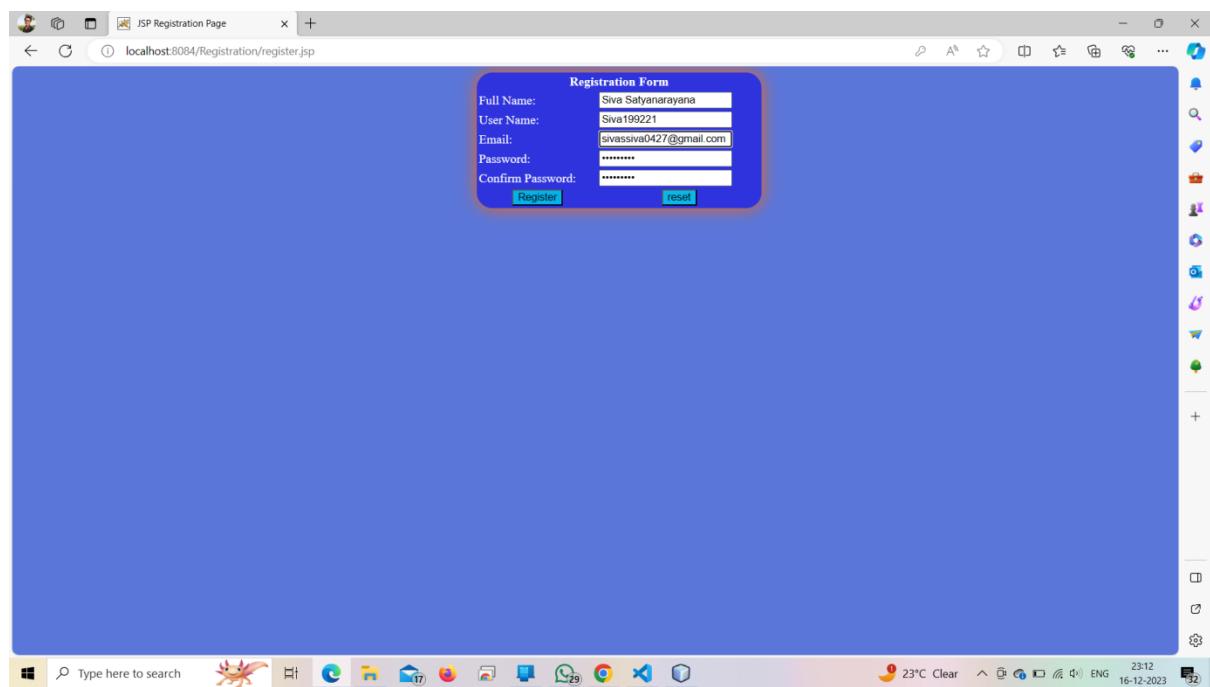
```
alter table `registration`.`register` auto_increment = 1;
```

OUTPUTS:

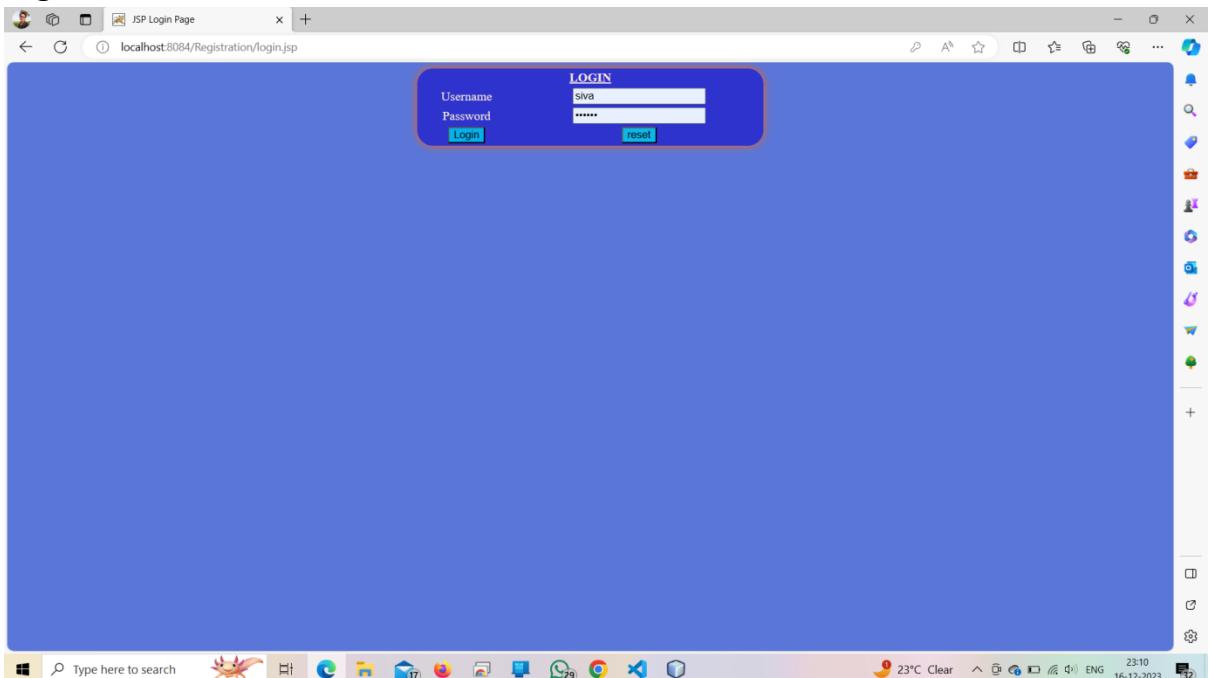
1. **Landing Page** consists of Navigation bar containing components like **HOME, ABOUT, CONTACT, OUR SERVICES, LOGIN AND REGSITER** buttons as shown below:



2. **User** has to be registered at first, if he/she wanted to login as shown below:



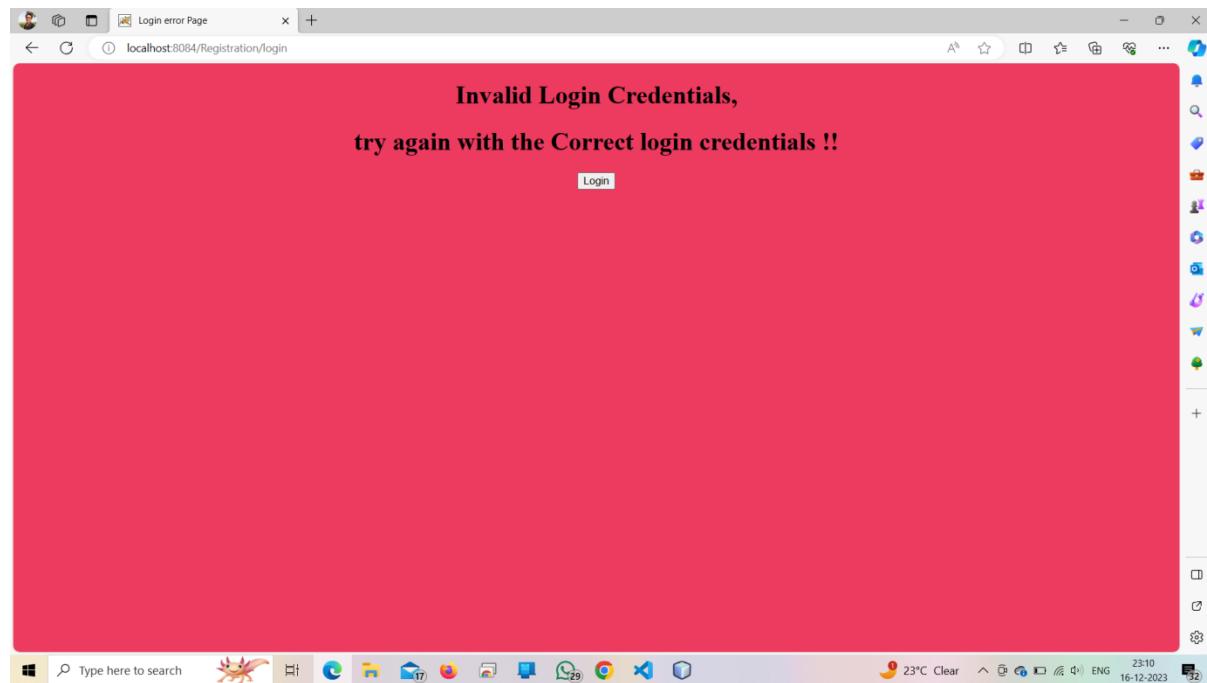
- 3.** Upon successful registration, the User can login into his/her account with registered Username and Password as shown below:



4.



- 5.** If User not yet created account, trying to login and authenticate into his/her account, the user gets an error as shown below:



MAIN OUTPUT:-

MySQL Query Browser - root@localhost:3306 / registration

File Edit View Query Script Tools Window Help

SELECT * FROM register;

Resultset 1

	id	fullname	username	email	password	confirmpassword
1	3	siva salyanarayana	siva11421	sivasiva123@gmail.com	Apple@990	Apple@990
2	4	siva	salyanarayana	asiv332@gmail.com	Bat@1234	Bat@1234
3	5	Siva Salyanarayana	Siva199221	sivasiva0427@gmail.com	123456789	123456789

3 rows fetched in 0.011s (0.000s)

Execute Stop

Schemas

- information_schema
- mysql
- registration
- register
- test

Syntax Functions

- Data Manipulation...
- Data Definition
- MySQL Utility
- Transactional ...

23°C Clear 23:13 16-12-2023

Type here to search

CONCLUSION:-

Java Full stack development deals with the end-to-end development of applications. It includes both the frontend and backend development of an application. The front end is usually accessed by a client, and the back end performs all the business logic and also stores the data.

During this internship, i learned all the basic Java concepts and implemented them in the form of daily assignments. Later i also learned the advanced Java concepts and practiced them. Finally in the form of a mini project, I have developed a small application which sends the data to the backend database and also does the client side validation by using NetBeans.

In future i am looking forward to learning the other full stack technologies and implementing them to gain more knowledge.