"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editio

Addition: 15 Subtraction: 5 Multiplication: 50 Division: 2

Process finished with exit code θ

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.

Rectangle Area: 15.0

Circle Area: 50.26548245743669

Triangle Area: 6.0

Process finished with exit code 0

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\

Make: Volvo, Model: FH16, Year: 2020, Fuel: Diesel

Fuel Efficiency: 8.0 km/l Distance with 20L: 160.0 km

Max Speed: 120 km/h

Make: Toyota, Model: Corolla, Year: 2021, Fuel: Petrol

Fuel Efficiency: 15.0 km/l Distance with 20L: 300.0 km

Max Speed: 180 km/h

Make: Yamaha, Model: R15, Year: 2022, Fuel: Petrol

Fuel Efficiency: 35.0 km/l Distance with 20L: 700.0 km

Max Speed: 160 km/h

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains

Rectangle Area: 15.0

Circle Area: 50.26548245743669

Triangle Area: 6.0

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains'

File does not exist.

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community File created: C:\Users\atulm\Desktop\Atullya.txt

Data written to: C:\Users\atulm\Desktop\Atullya.txt

Reading file content:

Hello, my name is Atullya.

This file is written and read using FileWriter and FileReader.

Process finished with exit code 0



```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA File Name: sample.txt
Absolute Path: C:\Users\atulm\Desktop\JDBC\sample.txt
Size (bytes): 0
Readable: true
Writable: true
Executable: true
```

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Atullya is trying to withdraw $70
Atullya is withdrawing...
Atullya withdrew successfully. Remaining balance: $30
Avaya is trying to withdraw $50
Avaya can't withdraw. Not enough balance!
```

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1\ Odd: 1 Even: 2
Odd: 3 Even: 4
Odd: 5 Even: 6
Odd: 7 Even: 8
Odd: 9 Even: 10
Odd: 11 Even: 12
Odd: 13 Even: 14
Odd: 15 Even: 16
Odd: 17 Even: 18
Odd: 17 Even: 18
Odd: 19 Even: 20
```

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
DepositThread-1 deposited: 500 | Balance: 5500
DepositThread-2 deposited: 500 | Balance: 6000
WithdrawThread-2 withdrew: 700 | Balance: 5300
WithdrawThread-1 withdrew: 700 | Balance: 4600
```

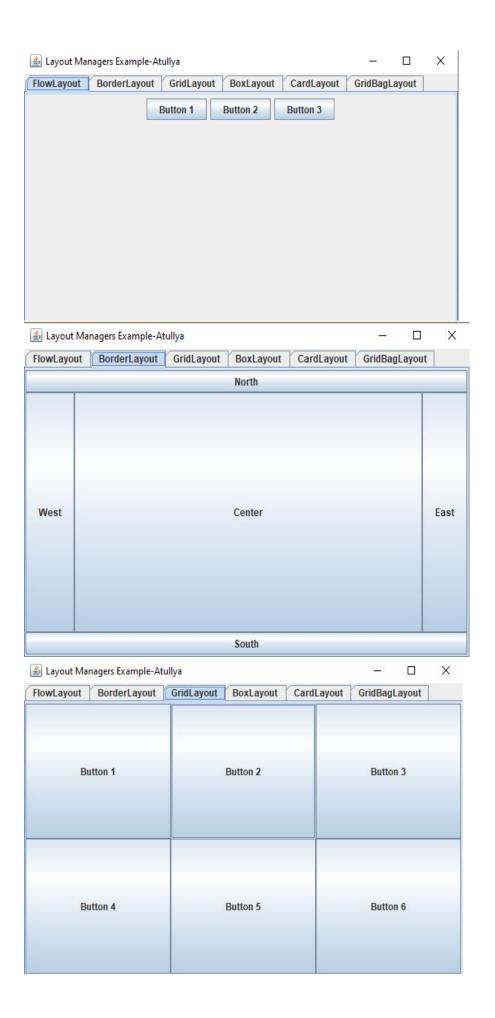
LAB 7:- Graphical User Interface(GUI) Programming in Java

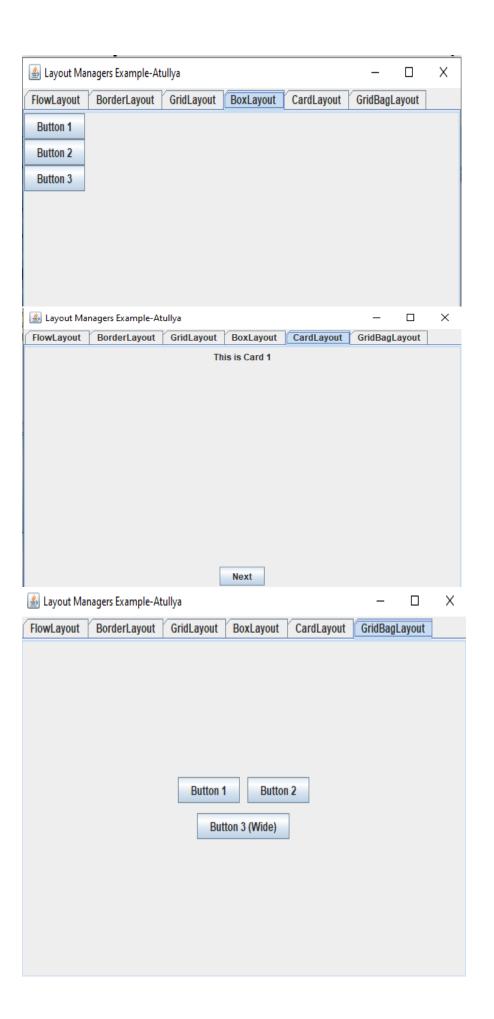
- a. Write a Java Swing application that demonstrates the use of all common layout managers in Swing. Your program should include examples of the following layouts:
- FlowLayout
- BorderLayout
- GridLayout
- BoxLayout
- CardLayout
- GridBagLayout

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class LayoutExample extends JFrame {
  CardLayout cardLayout;
  JPanel cardPanel;
  public LayoutExample() {
    setTitle("Layout Managers Example-Atullya");
    setSize(600, 400);
    setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
    setLocationRelativeTo(null);
    JTabbedPane tabbedPane = new JTabbedPane();
    JPanel flowPanel = new JPanel(new FlowLayout());
    flowPanel.add(new JButton("Button 1"));
    flowPanel.add(new JButton("Button 2"));
    flowPanel.add(new JButton("Button 3"));
    tabbedPane.add("FlowLayout", flowPanel);
    JPanel borderPanel = new JPanel(new BorderLayout());
    borderPanel.add(new JButton("North"), BorderLayout.NORTH);
    borderPanel.add(new JButton("South"), BorderLayout.SOUTH);
    borderPanel.add(new JButton("East"), BorderLayout. EAST);
    borderPanel.add(new JButton("West"), BorderLayout. WEST);
    borderPanel.add(new JButton("Center"), BorderLayout. CENTER);
    tabbedPane.add("BorderLayout", borderPanel);
    JPanel gridPanel = new JPanel(new GridLayout(2, 3));
    for (int i = 1; i \le 6; i++) {
       gridPanel.add(new JButton("Button" + i));
    tabbedPane.add("GridLayout", gridPanel);
    JPanel boxPanel = new JPanel();
    boxPanel.setLayout(new BoxLayout(boxPanel, BoxLayout.Y AXIS));
    boxPanel.add(new JButton("Button 1"));
    boxPanel.add(new JButton("Button 2"));
    boxPanel.add(new JButton("Button 3"));
    tabbedPane.add("BoxLayout", boxPanel);
    cardLayout = new CardLayout();
```

```
cardPanel = new JPanel(cardLayout);
  JPanel card1 = new JPanel();
  card1.add(new JLabel("This is Card 1"));
  JPanel card2 = new JPanel();
  card2.add(new JLabel("This is Card 2"));
  cardPanel.add(card1, "Card1");
  cardPanel.add(card2, "Card2");
  JPanel cardButtonPanel = new JPanel();
  JButton nextButton = new JButton("Next");
  cardButtonPanel.add(nextButton);
  nextButton.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) {
       cardLayout.next(cardPanel);
  });
  JPanel cardLayoutPanel = new JPanel(new BorderLayout());
  cardLayoutPanel.add(cardPanel, BorderLayout.CENTER);
  cardLayoutPanel.add(cardButtonPanel, BorderLayout.SOUTH);
  tabbedPane.add("CardLayout", cardLayoutPanel);
  JPanel gridBagPanel = new JPanel(new GridBagLayout());
  GridBagConstraints gbc = new GridBagConstraints();
  gbc.insets = new Insets(5, 5, 5, 5);
  gbc.gridx = 0; gbc.gridy = 0;
  gridBagPanel.add(new JButton("Button 1"), gbc);
  gbc.gridx = 1; gbc.gridy = 0;
  gridBagPanel.add(new JButton("Button 2"), gbc);
  gbc.gridx = 0; gbc.gridy = 1;
  gbc.gridwidth = 2;
  gridBagPanel.add(new JButton("Button 3 (Wide)"), gbc);
  tabbedPane.add("GridBagLayout", gridBagPanel);
  add(tabbedPane);
  setVisible(true);
public static void main(String[] args) {
  new LayoutExample();
```

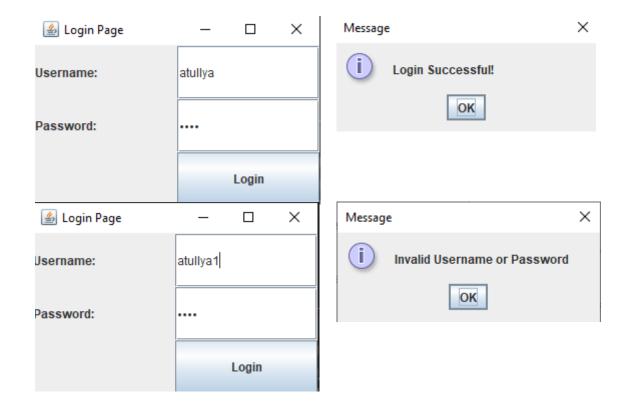
}





b. Create a login page in Swing and add events to it on click of login button

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class LoginExample extends JFrame implements ActionListener {
  JTextField userField;
  JPasswordField passField;
  JButton loginButton;
  public LoginExample() {
    setTitle("Login Page");
    setSize(300, 200);
    setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
    setLayout(new GridLayout(3, 2));
    JLabel userLabel = new JLabel("Username:");
    JLabel passLabel = new JLabel("Password:");
    userField = new JTextField();
    passField = new JPasswordField();
    loginButton = new JButton("Login");
    add(userLabel);
    add(userField);
    add(passLabel);
    add(passField);
    add(new JLabel());
    add(loginButton);
    loginButton.addActionListener(this);
    setLocationRelativeTo(null);
    setVisible(true);
  public void actionPerformed(ActionEvent e) {
    String username = userField.getText();
    String password = new String(passField.getPassword());
    if (username.equals("atullya") && password.equals("1234")) {
       JOptionPane.showMessageDialog(this, "Login Successful!");
    } else {
       JOptionPane.showMessageDialog(this, "Invalid Username or Password");
  public static void main(String[] args) {
    new LoginExample();
}
```



Lab 8:- JDBC

Develop a Java application using JDBC to perform all four CRUD operations —Create, Read, Update, and Delete — on a MySQL database named student, assuming that the student table has six attributes (ID, name, age, address, email, and phone). [Make an email and phone number unique]

```
import java.sql.*;
import java.util.Scanner;
public class SqlConnection {
  static final String URL = "jdbc:mysql://localhost:3307/student";
  static final String USER = "root";
  static final String PASSWORD = "zlozower123@z";
  Connection conn;
  Scanner sc = new Scanner(System.in);
  public SqlConnection() {
    try {
       Class.forName("com.mysql.cj.jdbc.Driver");
       conn = DriverManager.getConnection(URL, USER, PASSWORD);
       System.out.println("Connection successful!");
     } catch (ClassNotFoundException e) {
       System.out.println("MySQL JDBC Driver not found");
       e.printStackTrace();
     } catch (SQLException e) {
       System.out.println("Connection failed!");
       e.printStackTrace();
  }
  public void createStudent() {
    try {
       System.out.print("Enter Name: ");
       String name = sc.nextLine();
       System.out.print("Enter Age: ");
       int age = sc.nextInt();
       sc.nextLine();
       System.out.print("Enter Address: ");
       String address = sc.nextLine();
       System.out.print("Enter Email: ");
       String email = sc.nextLine();
       System.out.print("Enter Phone: ");
       String phone = sc.nextLine();
       String sql = "INSERT INTO student (name, age, address, email, phone) VALUES (?, ?, ?, ?,
?)";
       PreparedStatement ps = conn.prepareStatement(sql);
       ps.setString(1, name);
       ps.setInt(2, age);
       ps.setString(3, address);
       ps.setString(4, email);
       ps.setString(5, phone);
```

```
ps.executeUpdate();
     System.out.println("Student inserted successfully!");
  } catch (SQLIntegrityConstraintViolationException e) {
     System.out.println("Error: Email or Phone must be unique!");
  } catch (SQLException e) {
     e.printStackTrace();
}
public void readStudents() {
  try {
     String sql = "SELECT * FROM student";
     Statement stmt = conn.createStatement();
     ResultSet rs = stmt.executeQuery(sql);
     System.out.println("ID | Name | Age | Address | Email | Phone");
     while (rs.next()) {
       System.out.println(rs.getInt("id") + " | " +
            rs.getString("name") + " | " +
            rs.getInt("age") + " | " +
            rs.getString("address") + " | " +
            rs.getString("email") + " | " +
            rs.getString("phone"));
  } catch (SQLException e) {
     e.printStackTrace();
}
public void updateStudent() {
  try {
     System.out.print("Enter Student ID to update: ");
     int id = sc.nextInt();
     sc.nextLine();
     System.out.print("Enter new Address: ");
     String address = sc.nextLine();
     System.out.print("Enter new Phone: ");
     String phone = sc.nextLine();
     String sql = "UPDATE student SET address=?, phone=? WHERE id=?";
     PreparedStatement ps = conn.prepareStatement(sql);
     ps.setString(1, address);
     ps.setString(2, phone);
     ps.setInt(3, id);
     int rows = ps.executeUpdate();
     if (rows > 0)
       System.out.println("Student updated successfully!");
     else
       System.out.println("Student ID not found.");
  } catch (SQLIntegrityConstraintViolationException e) {
     System.out.println("Error: Phone must be unique!");
```

```
} catch (SQLException e) {
     e.printStackTrace();
}
public void deleteStudent() {
  try {
     System.out.print("Enter Student ID to delete: ");
     int id = sc.nextInt();
     String sql = "DELETE FROM student WHERE id=?";
     PreparedStatement ps = conn.prepareStatement(sql);
     ps.setInt(1, id);
     int rows = ps.executeUpdate();
     if (rows > 0)
       System.out.println("Student deleted successfully!");
       System.out.println("Student ID not found.");
   } catch (SQLException e) {
     e.printStackTrace();
}
public void menu() {
  while (true) {
     System.out.println("\n--- Student CRUD Menu ---");
     System.out.println("1. Create Student");
     System.out.println("2. Read Students");
     System.out.println("3. Update Student");
     System.out.println("4. Delete Student");
     System.out.println("5. Exit");
     System.out.print("Choose an option: ");
     int choice = sc.nextInt();
     sc.nextLine();
     switch (choice) {
       case 1: createStudent(); break;
       case 2: readStudents(); break;
       case 3: updateStudent(); break;
       case 4: deleteStudent(); break;
       case 5: System.out.println("Exiting..."); return;
       default: System.out.println("Invalid choice!");
     }
   }
public static void main(String[] args) {
  SqlConnection app = new SqlConnection();
  app.menu();
}
```

}

- ō X MvSOL 8.0 Command Line Client Enter password: ********** Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 10 Server version: 8.0.42 MySQL Community Server - GPL Copyright (c) 2000, 2025, Oracle and/or its affiliates. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective Type 'help;' or '\h' for help. Type '\c' to clear the current input statement. mysql> CREATE DATABASE student; Query OK, 1 row affected (0.04 sec) mysql> USE student; age INT NOT NULL, address VARCHAR(255), email VARCHAR(100) UNIQUE, phone VARCHAR(20) UNIQUE Query OK, 0 rows affected (0.78 sec) ysql> "C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition Connection successful! --- Student CRUD Menu ---1. Create Student 2. Read Students 3. Update Student

```
4. Delete Student
5. Exit
Choose an option: 1
Enter Name: Atullya
Enter Age: 22
Enter Address: Bafal
Enter Email: atullya11@gmail.com
Enter Phone: 1234567890
Student inserted successfully!
--- Student CRUD Menu ---
1. Create Student
2. Read Students
3. Update Student
4. Delete Student
5. Exit
Choose an option: 2
ID | Name | Age | Address | Email | Phone
1 | Atullya | 22 | Bafal | atullya11@gmail.com | 1234567890
```

```
--- Student CRUD Menu ---
1. Create Student
2. Read Students
3. Update Student
4. Delete Student
5. Exit
Choose an option: 3
Enter Student ID to update: 1
Enter new Address: Nagarjun-4
Enter new Phone: 9813324333
Student updated successfully!
--- Student CRUD Menu ---
1. Create Student
2. Read Students
3. Update Student
4. Delete Student
5. Exit
Choose an option: 2
ID | Name | Age | Address | Email | Phone
1 | Atullya | 22 | Nagarjun-4 | atullya11@gmail.com | 9813324333
--- Student CRUD Menu ---
1. Create Student
2. Read Students
3. Update Student
4. Delete Student
5. Exit
Choose an option: 4
Enter Student ID to delete: 1
Student deleted successfully!
```

Lab 9- Java Networking

- a. Write a Java program to:
- Create a URL object with the following address:

https://www.example.com:443/page/index.html?user=admin&ref=123

• Display the following components of the URL using appropriate methods:

```
- Protocol
- Host
- Port
- Path
- File
- Query
import java.net.*;
public class Lab09URLExample {
  public static void main(String[] args) {
     try {
       URL url = new
URL("https://www.example.com:443/page/index.html?user=admin&ref=123");
       System.out.println("Protocol: " + url.getProtocol());
       System.out.println("Host: " + url.getHost());
       System.out.println("Port: " + url.getPort());
       System.out.println("Path: " + url.getPath());
       System.out.println("File: " + url.getFile());
       System.out.println("Query: " + url.getQuery());
     } catch (MalformedURLException e) {
       System.out.println("Invalid URL!");
       e.printStackTrace();
  }
}
  C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community"
 Protocol: https
 Port: 443
 Path: /page/index.html
 File: /page/index.html?user=admin&ref=123
 Query: user=admin&ref=123
```

b. Write down the steps of creating Client and Server using Socket Programming

er.java

```
import java.net.ServerSocket;
import java.net.Socket;
import java.io.*;
public class Server {
  public static void main(String[] args) {
    int port = 5000;
    try (ServerSocket serverSocket = new ServerSocket(port)) {
       System.out.println("Server started on port " + port + ". Waiting for client...");
       Socket clientSocket = serverSocket.accept(); // accept client connection
       System.out.println("Client connected: " + clientSocket.getInetAddress());
       BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
       PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
       String message;
       while ((message = in.readLine()) != null) {
          System.out.println("Client says: " + message);
         if (message.equalsIgnoreCase("bye-bye")) {
            out.println("Goodbye!");
            break:
         out.println("Server received: " + message);
       in.close():
       out.close();
       clientSocket.close();
       System.out.println("Connection closed.");
     } catch (IOException e) {
       e.printStackTrace();
}
Client.java
import java.net.Socket;
import java.io.*;
import java.util.Scanner;
public class Client {
  public static void main(String[] args) {
    String host = "localhost"; // server IP
    int port = 5000;
    try (Socket socket = new Socket(host, port);
        BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        Scanner sc = new Scanner(System.in)) {
       System.out.println("Connected to server at " + host + ":" + port);
       String message;
       while (true) {
          System.out.print("Enter message: ");
```

```
message = sc.nextLine();
out.println(message); // send to server
String response = in.readLine(); // read server response
System.out.println("Server: " + response);

if (message.equalsIgnoreCase("bye-bye")) {
    break;
}
} catch (IOException e) {
    e.printStackTrace();
}
}
```

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ
Connected to server at localhost:5000
Enter message: Hello, It's me atullya
Server: Server received: Hello, It's me atullya

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\.
Server started on port 5000. Waiting for client...
Client connected: /127.0.0.1
Client says: Hello, It's me atullya
```

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Connected to server at localhost:5000
Enter message: Hello, It's me atullya
Server: Server received: Hello, It's me atullya
Enter message: How are you?
Server: Server received: How are you?

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Server started on port 5000. Waiting for client...
Client connected: /127.0.0.1
Client says: Hello, It's me atullya
Client says: How are you?
```

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program

Connected to server at localhost:5000 Enter message: Hello, It's me atullya

Server: Server received: Hello, It's me atullya

Enter message: How are you?

Server: Server received: How are you?

Enter message: bye-bye

Server: Goodbye!

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program

Server started on port 5000. Waiting for client...

Client connected: /127.0.0.1

Client says: Hello, It's me atullya

Client says: How are you?

Client says: bye-bye Connection closed.

Lab 10:-JSP

Create a login form in login.jsp, using Servlet authenticate (accept POST request) the username (as admin) and password (as nccs#123), forward to dashboard.jsp if login is successful and include error message if login fails in same login.jsp.

index.jsp

```
<%@ page contentType="text/html;charset=UTF-8" %>
<!DOCTYPE html>
<html>
<head>
  <title>Login Page</title>
</head>
<body>
  <h2>Login</h2>
  <form action="Login" method="post">
    <label>Username: </label>
    <input type="text" name="username" required><br><br>
    <label>Password: </label>
    <input type="password" name="password" required><br><br>
    <input type="submit" value="Login">
  </form>
</body>
</html>
Dashboard.jsp
<%@ page contentType="text/html;charset=UTF-8" %>
<!DOCTYPE html>
<html>
<head>
  <title>Dashboard</title>
</head>
<body>
  <h2>Welcome to Dashboard!</h2>
  You have successfully logged in.
</body>
</html>
LoginServlet.java
package com.example;
import java.io.IOException;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
```

public class LoginServlet extends HttpServlet {

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
       throws ServletException, IOException {
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    // Check credentials
    if ("admin".equals(username) && "nccs#123".equals(password)) {
       // Forward to dashboard.jsp
       RequestDispatcher dispatcher = request.getRequestDispatcher("dashboard.jsp");
       dispatcher.forward(request, response);
    } else {
       // Send back to index.jsp with error message
       request.setAttribute("errorMessage", "Invalid Username or Password!");
       RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");
       dispatcher.forward(request, response);
    }}}
\langle \rangle 6
                       i localhost:8080/jspServlet/
Login
Username:
Password:
Login
   > C
                            i localhost:8080/jspServlet/
Login
Username: admin
Password: .....
Login
                                        П
                                                 localhost:8080/jspServlet/Login
```

Welcome to Dashboard!

You have successfully logged in.

Lab 11:- Servlet

a. Servlet Program example in Java.

```
LoginExampleServlet.java
```

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class HelloServlet extends HttpServlet {
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
       throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<h2>Welcome to Servlet Example!</h2>");
    out.println("This is a simple Hello World Servlet.");
    out.println("</body></html>");
  }}
```

П

i) localhost:8080/jspServlet/Hello



This is a simple Hello World Servlet.

b. Servlet Program to implement Cookies and Sessions

}}

```
SessionCookies.java
package com.example;
import java.io.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;
@WebServlet(name = "helloServlet", value = "/session-cookie")
public class SessionCookies extends HttpServlet {
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String cookieName = "user";
    String user = request.getParameter("name");
    if (user != null && !user.isEmpty()) {
       Cookie cookie = new Cookie(cookieName, user);
       cookie.setMaxAge(60 * 60); // 1 hour
       response.addCookie(cookie);
     } else {
       Cookie[] cookies = request.getCookies();
       if (cookies != null) {
         for (Cookie c : cookies) {
            if (cookieName.equals(c.getName())) {
              user = c.getValue();
              break;
            }}}
    if (user == null || user.isEmpty())
       user = "Guest";
    HttpSession session = request.getSession();
    Integer visitCount = (Integer) session.getAttribute("visitCount");
    if (visitCount == null)
       visitCount = 1;
    else
       visitCount++;
    session.setAttribute("visitCount", visitCount);
    out.println("<html><body>");
    out.println("<h1>Hello, " + user + "!</h1>");
    out.println("Session visit count: " + visitCount + "");
    out.println("<form method='get'>");
    out.println("Enter your name: <input type='text' name='name'/>");
    out.println("<input type='submit' value='Set Name'/>");
    out.println("</form>");
    out.println("</body></html>");
```

Hello, Atullya Maharjan!

Session visit count: 1

Enter your name: Atullya Set Name

i localhost:8080/jspServlet/session-cookie?n

Hello, Atullya!

Session visit count: 2

Enter your name: Atullya Set Name