



Vidyavardhini's College Of Engineering & Technology

NAME : CHETAN BHUYAL
DATE : 04/08/2025

ROLL NO. : 16
SUBJECT : FULL STACK JAVA

AIM :

Program on Implementing Generic and HTTP servlet.

Description:

This program implements a JSP login page with database validation using JDBC and a servlet that handles user login, addition, update, deletion, and logout operations dynamically.

CODE :

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

public class OnlineExamUI extends JFrame implements ActionListener {
    CardLayout card = new CardLayout();
    JPanel main = new JPanel(card);
    JTextField userField, newUser;
    JPasswordField passField, newPass;
    JLabel questionLabel, scoreLabel;
    JRadioButton[] options = new JRadioButton[4];
    ButtonGroup bg = new ButtonGroup();
    JButton nextBtn, submitBtn;
    int qIndex = 0, score = 0, correct = 0, wrong = 0;
```

```

String username;

Map<String, String> users = new HashMap<>();
String[][] quiz = {
    {"Which keyword is used for inheritance?", "extends", "super",
    "this", "import", "1"},
    {"Which is not OOP principle?", "Inheritance", "Polymorphism",
    "Encapsulation", "Compilation", "4"},
    {"Which package is imported by default?", "util", "io", "lang",
    "awt", "3"},
    {"Which collection is synchronized?", "ArrayList", "Vector",
    "TreeSet", "HashSet", "2"},
    {"What does JVM stand for?", "Java Virtual Machine", "Java Vendor
    Model", "Just Virtual Memory", "None", "1"}
};

OnlineExamUI() {
    setTitle("🎓 Java Online Exam System 🎮");
    setSize(500, 400);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    main.add(loginPanel(), "login");
    main.add(signupPanel(), "signup");
    main.add(examPanel(), "exam");
    main.add(resultPanel(), "result");
    add(main);
    card.show(main, "login");
    setVisible(true);
}

JPanel loginPanel() {
    JPanel p = new JPanel(new GridLayout(6,1,10,10));
    p.setBackground(new Color(250,250,255));
    JLabel title = new JLabel("🔒 Login to Exam",
SwingConstants.CENTER);
    title.setFont(new Font("Segoe UI", Font.BOLD, 18));
    userField = new JTextField();
    passField = new JPasswordField();
    JButton loginBtn = new JButton("Login ✅");
    JButton signupBtn = new JButton("Signup 📝");

```

```

        loginBtn.addActionListener(this);
signupBtn.addActionListener(this);

        p.add(title); p.add(new JLabel("Username:")); p.add(userField);
        p.add(new JLabel("Password:")); p.add(passField);
        JPanel btnPanel = new JPanel(); btnPanel.add(loginBtn);
btnPanel.add(signupBtn);
        p.add(btnPanel);
        return p;
    }

    JPanel signupPanel() {
        JPanel p = new JPanel(new GridLayout(5,1,10,10));
        p.setBackground(new Color(240,255,250));
        JLabel title = new JLabel("NEW Signup for Exam",
SwingConstants.CENTER);
        title.setFont(new Font("Segoe UI", Font.BOLD, 18));
        newUser = new JTextField(); newPass = new JPasswordField();
        JButton createBtn = new JButton("Create Account 🎉");
        JButton backBtn = new JButton("Back ⬅️");
        createBtn.addActionListener(this);
backBtn.addActionListener(this);

        p.add(title); p.add(new JLabel("Set Username:")); p.add(newUser);
        p.add(new JLabel("Set Password:")); p.add(newPass);
        JPanel b = new JPanel(); b.add(createBtn); b.add(backBtn);
        p.add(b);
        return p;
    }

    JPanel examPanel() {
        JPanel p = new JPanel(null);
        questionLabel = new JLabel("", SwingConstants.LEFT);
        questionLabel.setBounds(30, 30, 420, 30);
        questionLabel.setFont(new Font("Segoe UI", Font.BOLD, 16));
        p.add(questionLabel);
        int y = 80;
        for (int i=0; i<4; i++) {
            options[i] = new JRadioButton();
            bg.add(options[i]);
            options[i].setBounds(50, y, 400, 25);
            p.add(options[i]); y += 40;
        }
    }

```

```

    }
    nextBtn = new JButton("Next ➡");
    submitBtn = new JButton("Submit 🚩");
    nextBtn.setBounds(150, 260, 100, 30);
    submitBtn.setBounds(270, 260, 100, 30);
    nextBtn.addActionListener(this);
submitBtn.addActionListener(this);
    p.add(nextBtn); p.add(submitBtn);
    loadQuestion();
    return p;
}

JPanel resultPanel() {
    JPanel p = new JPanel(new GridLayout(5,1,10,10));
    p.setBackground(new Color(255,248,220));
    JLabel resTitle = new JLabel("📊 Exam Result",
SwingConstants.CENTER);
    resTitle.setFont(new Font("Segoe UI", Font.BOLD, 20));
    scoreLabel = new JLabel("", SwingConstants.CENTER);
    JButton retry = new JButton("Main Menu ↺");
    retry.addActionListener(this);
    p.add(resTitle); p.add(scoreLabel); p.add(new JLabel(""));
    p.add(retry);
    return p;
}

void loadQuestion() {
    if (qIndex < quiz.length) {
        questionLabel.setText("Q" + (qIndex+1) + ": " +
quiz[qIndex][0]);
        for (int i=0; i<4; i++) options[i].setText(quiz[qIndex][i+1]);
        bg.clearSelection();
    }
}

void calculate() {
    int ans = -1;
    for (int i=0; i<4; i++) if (options[i].isSelected()) ans = i+1;
    if (ans == Integer.parseInt(quiz[qIndex][5])) { score+=10;
correct++; }
}

```

```

        else wrong++;
        qIndex++;
        if (qIndex == quiz.length) card.show(main, "result");
        else loadQuestion();
    }



    void showResult() {
        String rank = (score >= 40) ? "🏆 GOLD" : (score >= 20) ? "🥈 SILVER"
: "🥉 BRONZE";
        scoreLabel.setText("<html>👤 "+username+"<br>✅ Correct:
"+correct+
        " | ❌ Wrong: "+wrong+"<br>📊 Score: "+score+
        "/50<br>🌟 Rank: "+rank+"</html>");
    }

    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        switch(cmd) {
            case "Login ✅" -> {
                String u = userField.getText();
                String p = new String(passField.getPassword());
                if(users.containsKey(u) && users.get(u).equals(p)) {
                    username = u; qIndex=0; score=correct=wrong=0;
                    loadQuestion(); card.show(main, "exam");
                    JOptionPane.showMessageDialog(this, "Welcome "+u+"
👤");
                } else JOptionPane.showMessageDialog(this, "❌ Invalid
Credentials!");
            }
            case "Signup 📝" -> card.show(main, "signup");
            case "Back ⬅️" -> card.show(main, "login");
            case "Create Account 🧑" -> {
                String u = newUser.getText(), p = new
String(newPass.getPassword());
                if(u.isEmpty() || p.isEmpty())
JOptionPane.showMessageDialog(this, "⚠️ Fill all fields!");
                else if(users.containsKey(u))
JOptionPane.showMessageDialog(this, "⚠️ Username exists!");

```

```
        else { users.put(u,p);  
JOptionPane.showMessageDialog(this,"✅ Account Created! Please Login.");  
card.show(main,"login"); }  
    }  
    case "Next ➡" -> calculate();  
    case "Submit 🏁" -> { calculate(); showResult();  
card.show(main,"result"); }  
    case "Main Menu 🔄" -> card.show(main,"login");  
    }  
    }  
  
    public static void main(String[] args) { new OnlineExamUI(); }  
}
```

OUTPUT:

 Java Online Exam System 

Login to Exam

Username:

Password:

CONCLUSION :

The program demonstrates a JSP login system with database validation using JDBC, where a single servlet handles login, add, update, delete, and logout operations. It ensures secure user management and displays dynamic success/error messages.