Assignment 2

Muhammad Usman

Sp23701

Numl-S23-26737

5th Semester

SCD

25 May-2025

Submitted to: Ms. Jaweria Kanwl

National University of Modern Languages

Islamabad

Github Repository link: https://github.com/Muhammad-Usman993/Online-Course-Enrollment-System--Project

1. Implementation

The project was implemented according to the design specifications outlined for an Online Course Enrollment System with user authentication. The system consists of three major components implemented in Java Swing:

- **Signup Module:**
  Allows new users to register with a unique username and password. User credentials are stored in a simple in-memory HashMap acting as a user database.
- **Login Module:**
  Registered users can log in with their credentials. On successful login, the user gains access to the enrollment system.
- **Enrollment Module:**
  Users can input student details (name, father's name, semester, program/department) and select a course from the available list. The system checks seat availability before enrolling the student.

The EnrollmentService class manages the enrollment logic ensuring course capacity limits are respected and both course and student enrollment lists are updated accordingly.

All UI panels (Signup, Login, Enrollment) are separated for better user flow and clarity, switching views dynamically within the main frame.

Courses and instructors are pre-populated during system initialization as per the requirements.

Key Design Highlights:

- User-friendly GUI with clear input validation and status messages.
- Separation of concerns: authentication and enrollment functionalities are modular.
- In-memory data structures used to keep track of users, students, and courses.
- Dynamic updating of UI components (e.g., course dropdown) reflecting current system state.

2. Unit Testing:

Unit tests were designed to cover major functionality areas:

User Authentication Tests:

- Verify signup rejects empty username or password.
- Verify signup rejects duplicate usernames.
- Verify login accepts correct credentials.
- Verify login rejects invalid username/password.

Enrollment Tests:

- Enrolling a student into a course with available seats should succeed.
- Enrolling a student into a course without seats should fail.
- Student enrollment records and course enrollment records must be consistent.

3. Test Results and Bug Reports

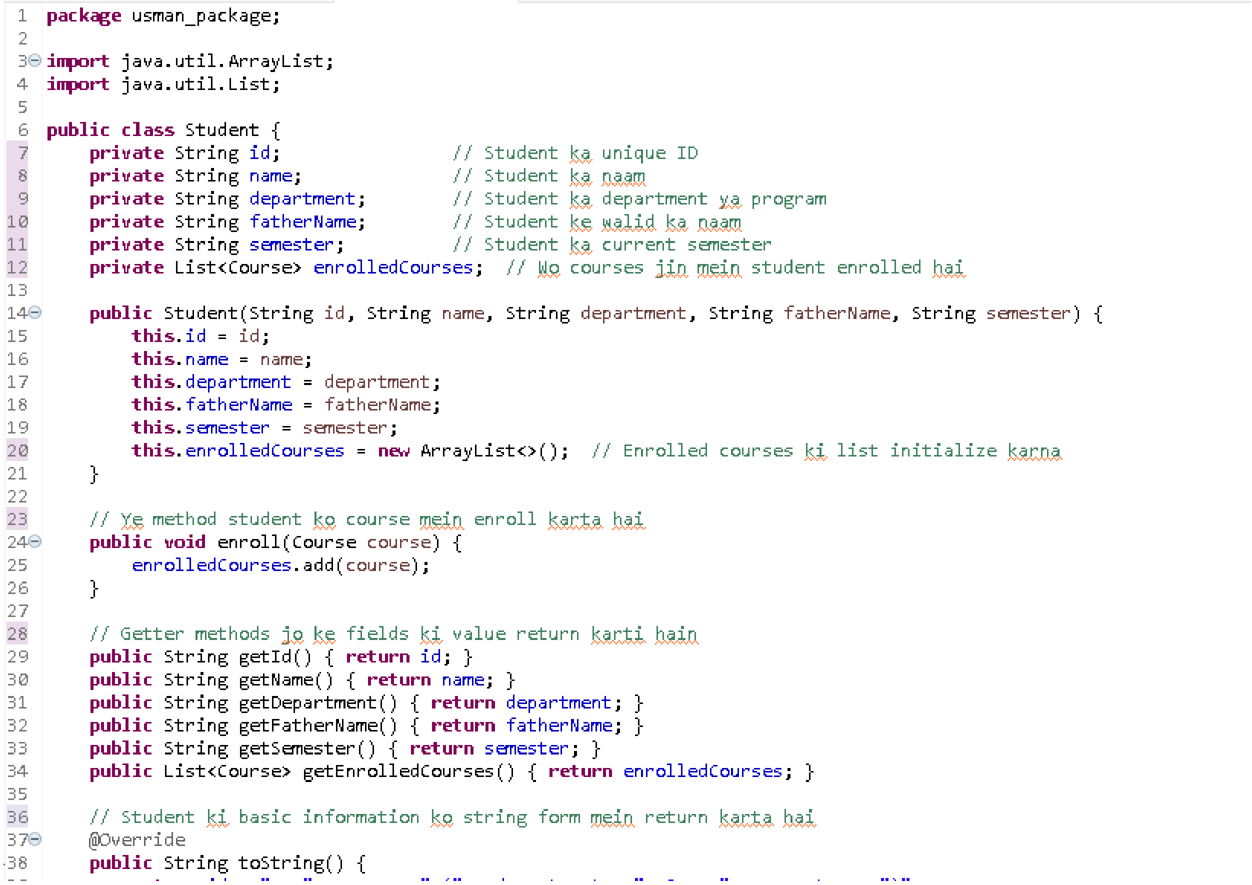
- **Bug #1: NullPointerException during student enrollment**
  Initially, the student object did not initialize the internal list to hold enrolled courses, causing an exception when adding courses.
  Fix: Properly initialize the list in the student constructor.
- **Bug #2: Duplicate usernames allowed during signup**
  The signup logic was fixed to check existing usernames before allowing registration.
- **Bug #3: Enrollment succeeded even when course was full**
  The seat availability check was improved in EnrollmentService.enroll() method.
- **UI Improvements:**
  Added clear error/success messages for signup/login failures and enrollment status, enhancing user experience.

No critical issues remain after fixes. The system works according to requirements.

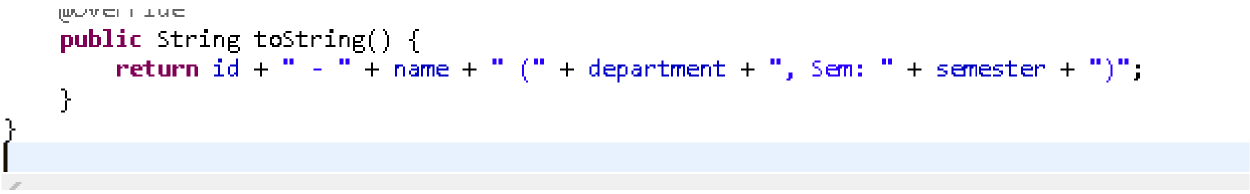
Documentation of Implementation:

The code is well-commented, with detailed explanations for classes and important methods.

Student.java:

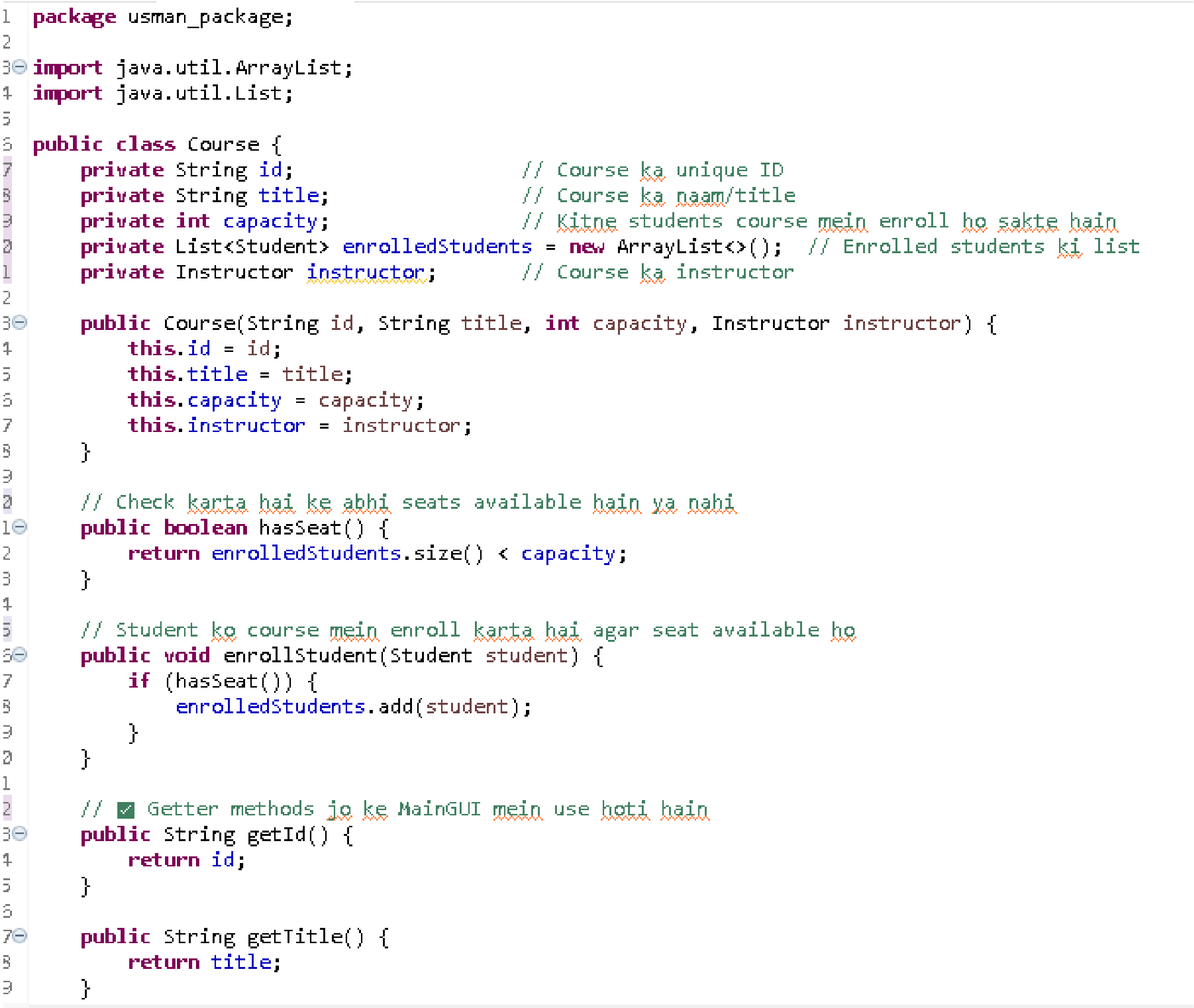


```java
package usman_package;

import java.util.ArrayList;
import java.util.List;

public class Student {
    private String id;                 // Student ka unique ID
    private String name;               // Student ka naam
    private String department;         // Student ka department ya program
    private String fatherName;         // Student ke walid ka naam
    private String semester;           // Student ka current semester
    private List<Course> enrolledCourses;  // Wo courses jin mein student enrolled hai

    public Student(String id, String name, String department, String fatherName, String semester) {
        this.id = id;
        this.name = name;
        this.department = department;
        this.fatherName = fatherName;
        this.semester = semester;
        this.enrolledCourses = new ArrayList<>();  // Enrolled courses ki list initialize karna
    }

    // Ye method student ko course mein enroll karta hai
    public void enroll(Course course) {
        enrolledCourses.add(course);
    }

    // Getter methods jo ke fields ki value return karti hain
    public String getId() { return id; }
    public String getName() { return name; }
    public String getDepartment() { return department; }
    public String getFatherName() { return fatherName; }
    public String getSemester() { return semester; }
    public List<Course> getEnrolledCourses() { return enrolledCourses; }

    // Student ki basic information ko string form mein return karta hai
    @Override
    public String toString() {
```



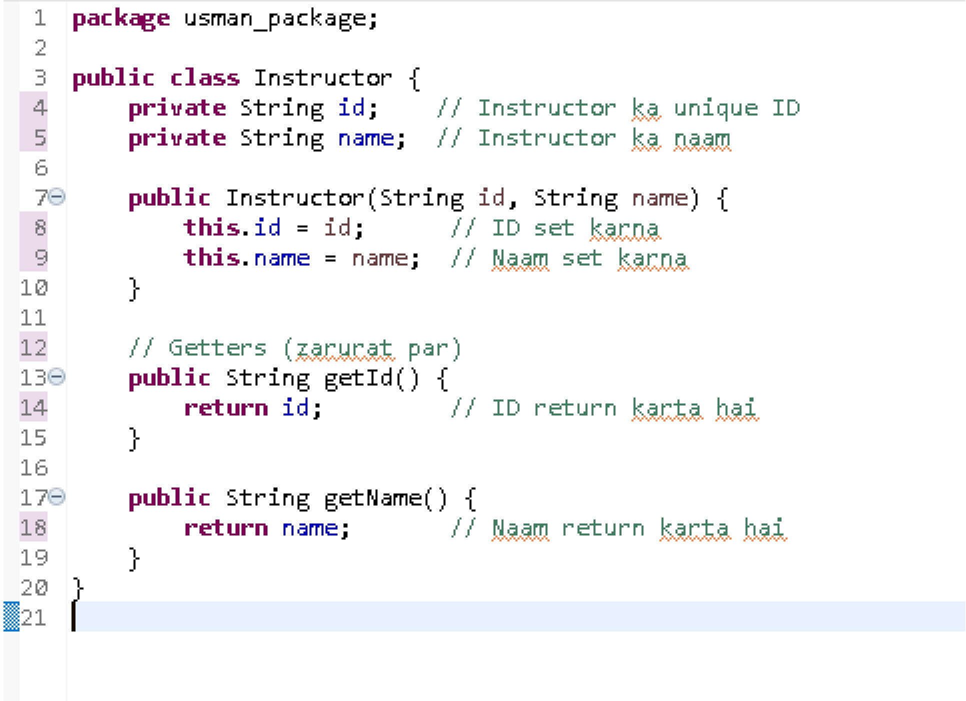
```java
    @Override
    public String toString() {
        return id + " - " + name + " (" + department + ", Sem: " + semester + ")";
    }
}
```

Course.java:



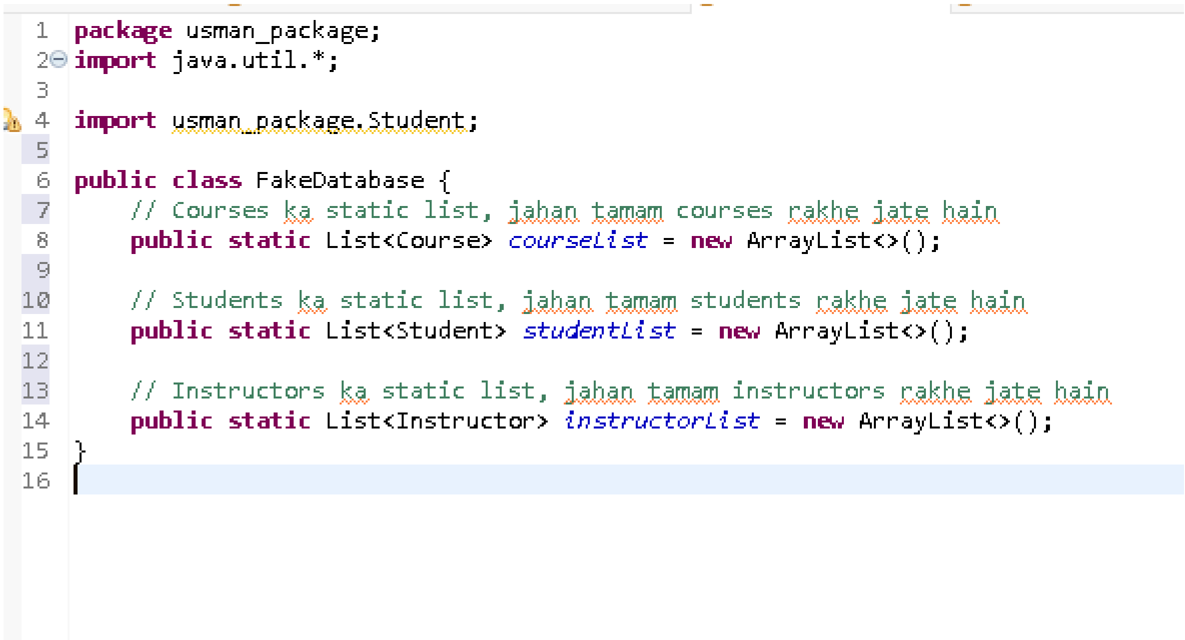
```java
package usman_package;

import java.util.ArrayList;
import java.util.List;

public class Course {
    private String id;                      // Course ka unique ID
    private String title;                   // Course ka naam/title
    private int capacity;                   // Kitne students course mein enroll ho sakte hain
    private List<Student> enrolledStudents = new ArrayList<>();  // Enrolled students ki list
    private Instructor instructor;          // Course ka instructor

    public Course(String id, String title, int capacity, Instructor instructor) {
        this.id = id;
        this.title = title;
        this.capacity = capacity;
        this.instructor = instructor;
    }

    // Check karta hai ke abhi seats available hain ya nahi
    public boolean hasSeat() {
        return enrolledStudents.size() < capacity;
    }

    // Student ko course mein enroll karta hai agar seat available ho
    public void enrollStudent(Student student) {
        if (hasSeat()) {
            enrolledStudents.add(student);
        }
    }

    // ✅ Getter methods jo ke MainGUI mein use hoti hain
    public String getId() {
        return id;
    }

    public String getTitle() {
        return title;
    }
```

Instructor.java:



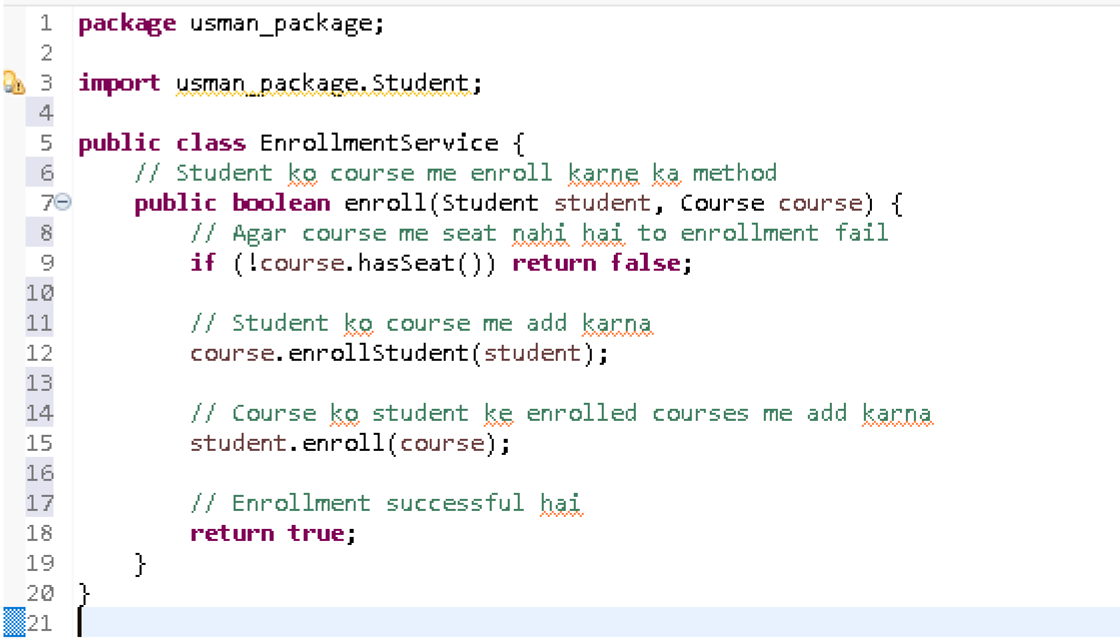
```java
package usman_package;

public class Instructor {
    private String id;    // Instructor ka unique ID
    private String name;  // Instructor ka naam

    public Instructor(String id, String name) {
        this.id = id;      // ID set karna
        this.name = name;  // Naam set karna
    }

    // Getters (zarurat par)
    public String getId() {
        return id;         // ID return karta hai
    }

    public String getName() {
        return name;       // Naam return karta hai
    }
}

```

FakeDatabse.java:



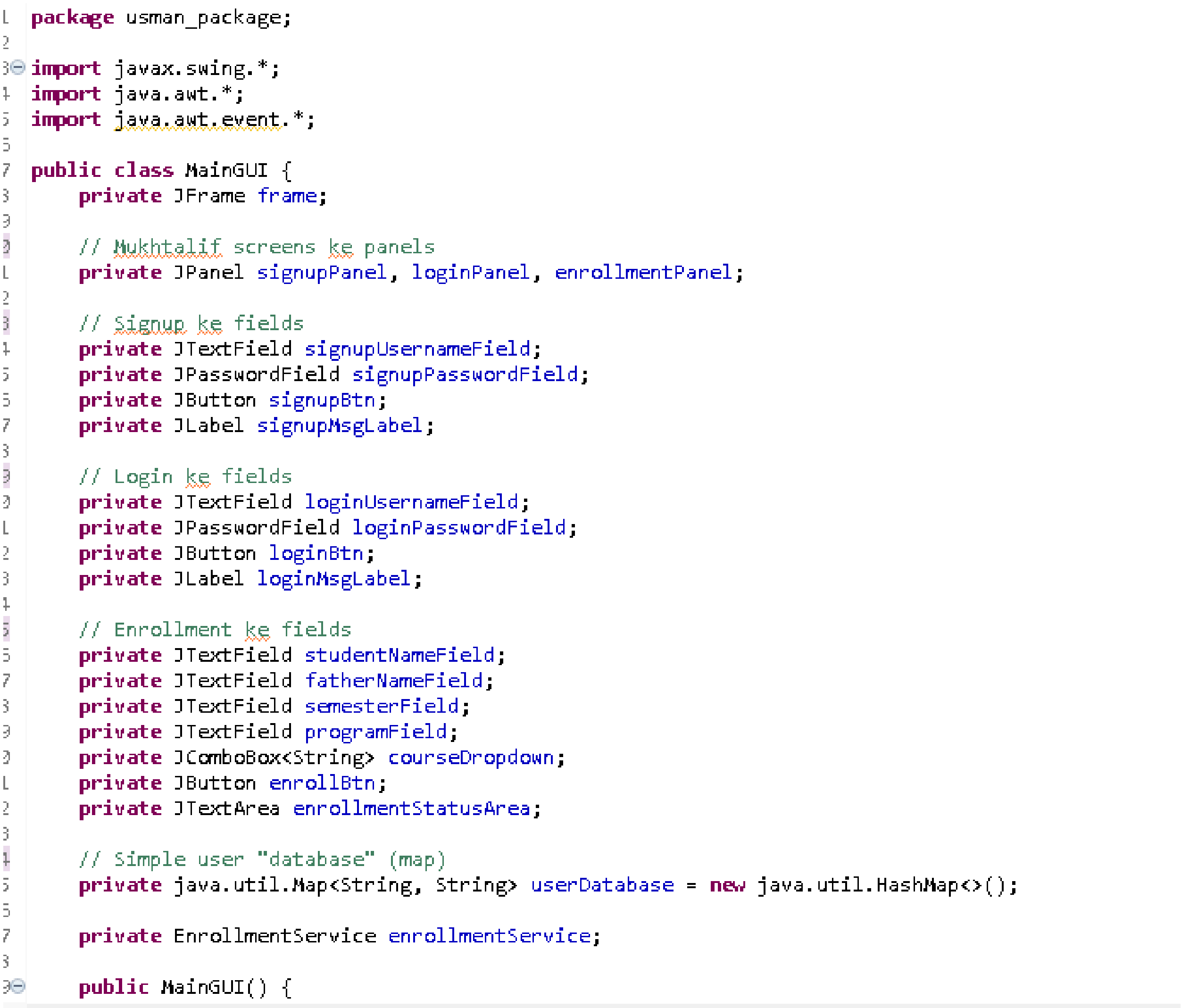
```java
package usman_package;
import java.util.*;

import usman_package.Student;

public class FakeDatabase {
    // Courses ka static list, jahan tamam courses rakhe jate hain
    public static List<Course> courseList = new ArrayList<>();

    // Students ka static list, jahan tamam students rakhe jate hain
    public static List<Student> studentList = new ArrayList<>();

    // Instructors ka static list, jahan tamam instructors rakhe jate hain
    public static List<Instructor> instructorList = new ArrayList<>();
}

```

Enrollments.java:

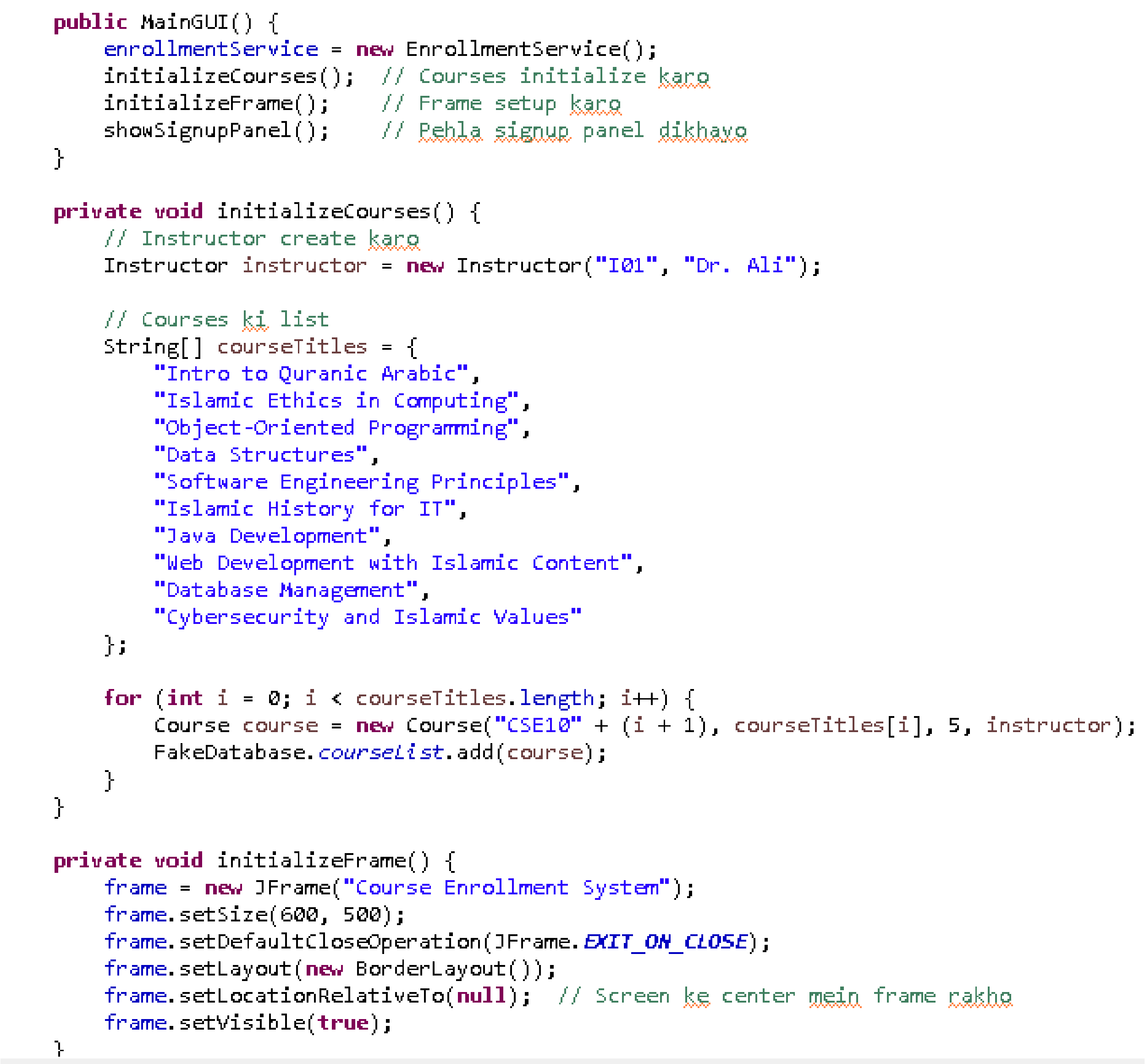


```java
package usman_package;

import usman_package.Student;

public class EnrollmentService {
    // Student ko course me enroll karne ka method
    public boolean enroll(Student student, Course course) {
        // Agar course me seat nahi hai to enrollment fail
        if (!course.hasSeat()) return false;

        // Student ko course me add karna
        course.enrollStudent(student);

        // Course ko student ke enrolled courses me add karna
        student.enroll(course);

        // Enrollment successful hai
        return true;
    }
}

```

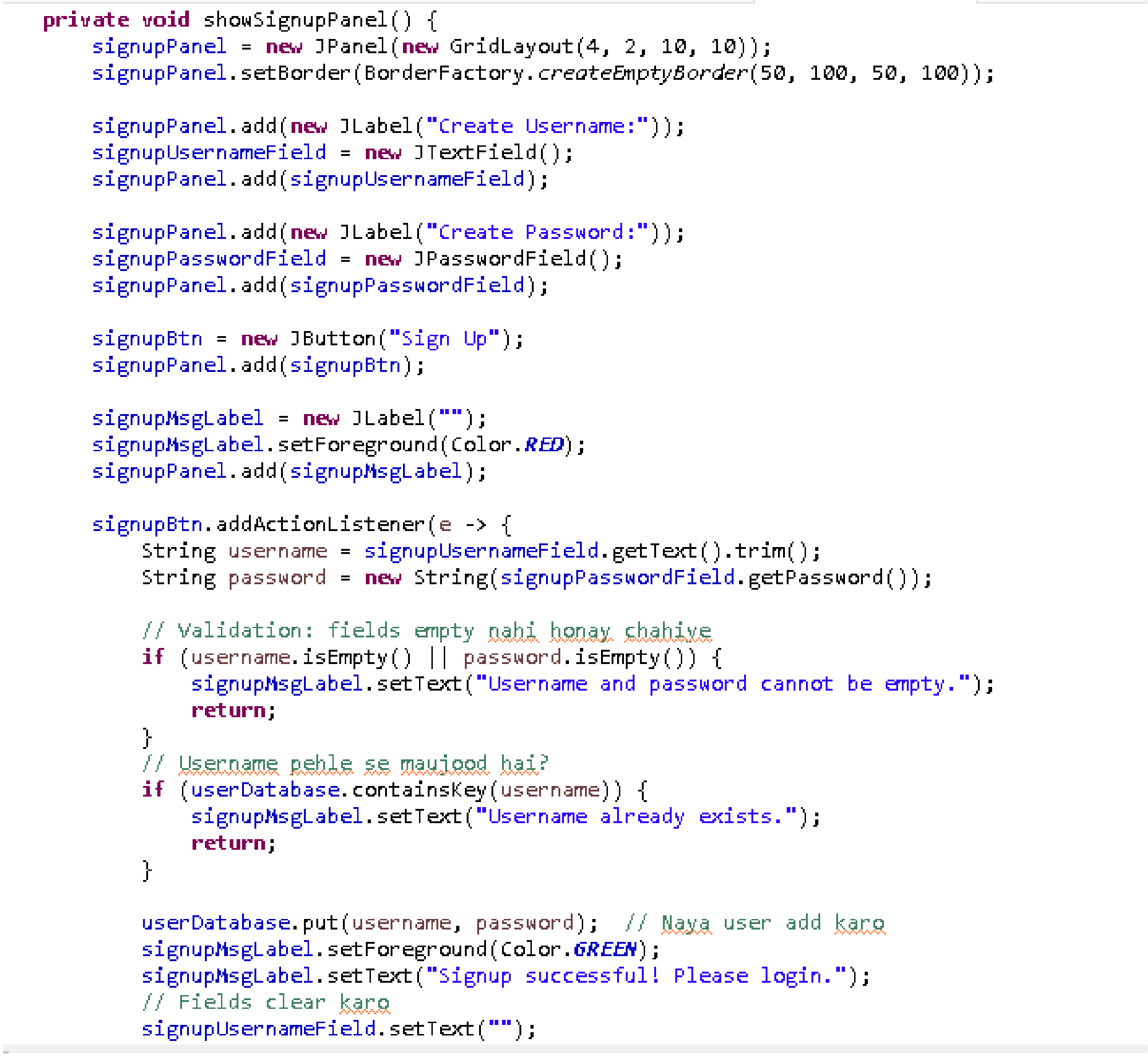
MainGUI.java:



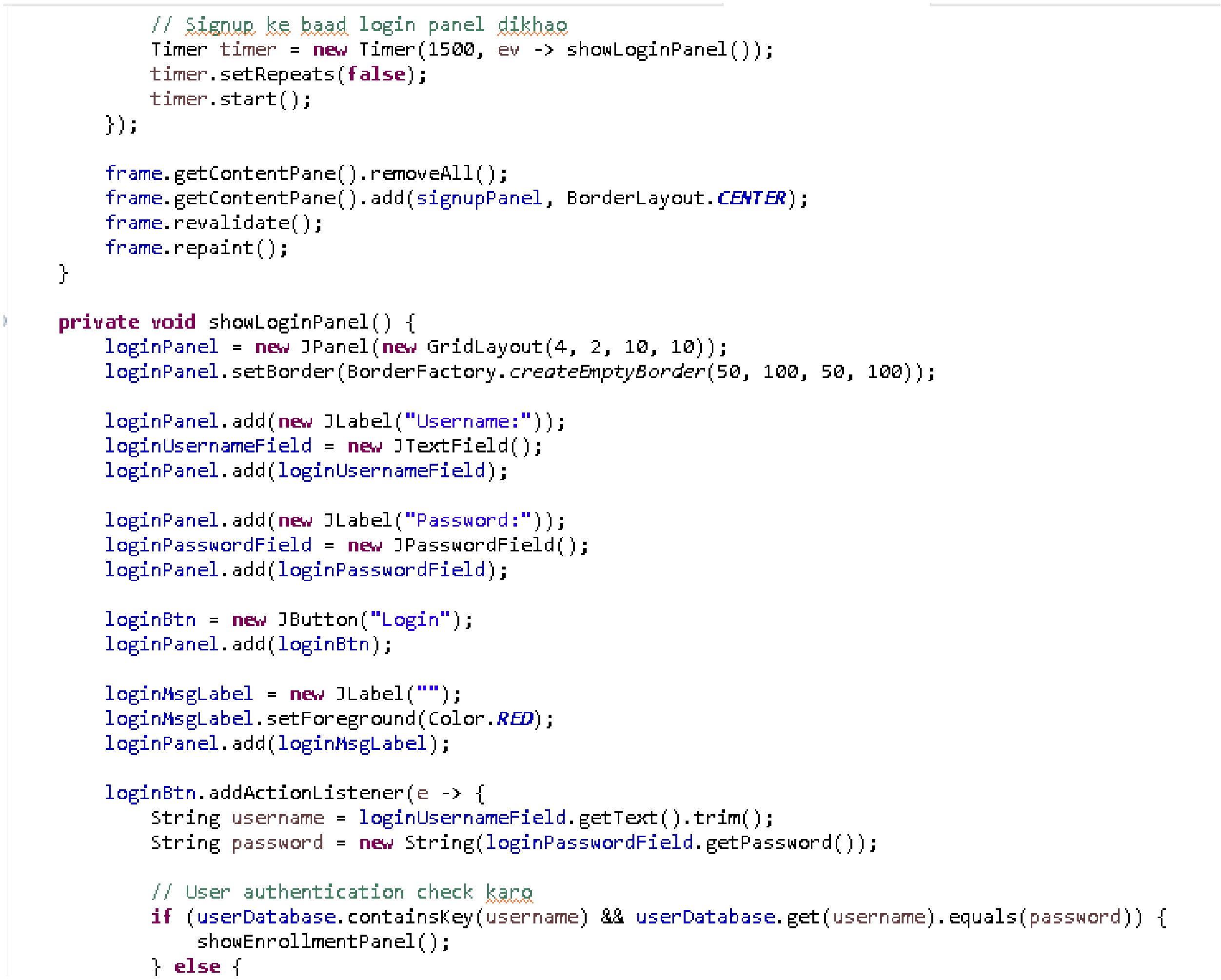
```java
package usman_package;

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

public class MainGUI {
    private JFrame frame;

    // Mukhtalif screens ke panels
    private JPanel signupPanel, loginPanel, enrollmentPanel;

    // Signup ke fields
    private JTextField signupUsernameField;
    private JPasswordField signupPasswordField;
    private JButton signupBtn;
    private JLabel signupMsgLabel;

    // Login ke fields
    private JTextField loginUsernameField;
    private JPasswordField loginPasswordField;
    private JButton loginBtn;
    private JLabel loginMsgLabel;

    // Enrollment ke fields
    private JTextField studentNameField;
    private JTextField fatherNameField;
    private JTextField semesterField;
    private JTextField programField;
    private JComboBox<String> courseDropdown;
    private JButton enrollBtn;
    private JTextArea enrollmentStatusArea;

    // Simple user "database" (map)
    private java.util.Map<String, String> userDatabase = new java.util.HashMap<>();

    private EnrollmentService enrollmentService;

    public MainGUI() {
```



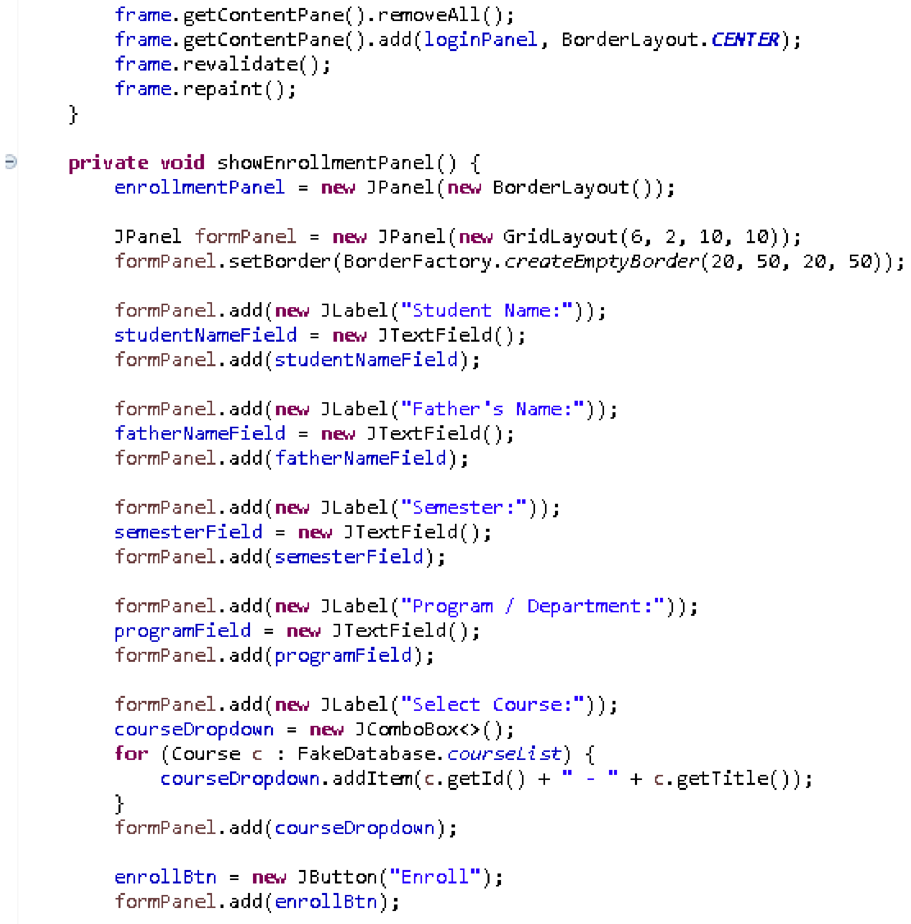
```
    public MainGUI() {
        enrollmentService = new EnrollmentService();
        initializeCourses();  // Courses initialize karo
        initializeFrame();    // Frame setup karo
        showSignupPanel();    // Pehla signup panel dikhayo
    }

    private void initializeCourses() {
        // Instructor create karo
        Instructor instructor = new Instructor("I01", "Dr. Ali");

        // Courses ki list
        String[] courseTitles = {
            "Intro to Quranic Arabic",
            "Islamic Ethics in Computing",
            "Object-Oriented Programming",
            "Data Structures",
            "Software Engineering Principles",
            "Islamic History for IT",
            "Java Development",
            "Web Development with Islamic Content",
            "Database Management",
            "Cybersecurity and Islamic Values"
        };

        for (int i = 0; i < courseTitles.length; i++) {
            Course course = new Course("CSE10" + (i + 1), courseTitles[i], 5, instructor);
            FakeDatabase.courseList.add(course);
        }
    }

    private void initializeFrame() {
        frame = new JFrame("Course Enrollment System");
        frame.setSize(600, 500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());
        frame.setLocationRelativeTo(null);  // Screen ke center mein frame rakho
        frame.setVisible(true);
    }
```



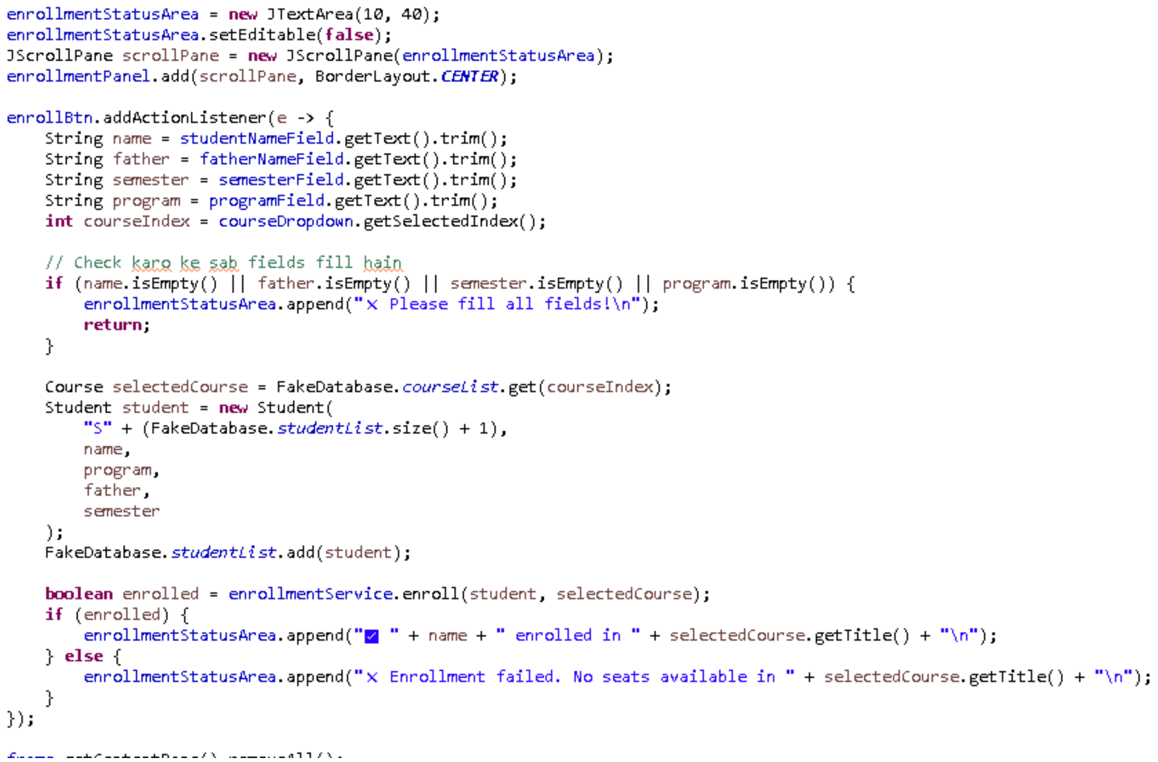
```
    private void showSignupPanel() {
        signupPanel = new JPanel(new GridLayout(4, 2, 10, 10));
        signupPanel.setBorder(BorderFactory.createEmptyBorder(50, 100, 50, 100));

        signupPanel.add(new JLabel("Create Username:"));
        signupUsernameField = new JTextField();
        signupPanel.add(signupUsernameField);

        signupPanel.add(new JLabel("Create Password:"));
        signupPasswordField = new JPasswordField();
        signupPanel.add(signupPasswordField);

        signupBtn = new JButton("Sign Up");
        signupPanel.add(signupBtn);

        signupMsgLabel = new JLabel("");
        signupMsgLabel.setForeground(Color.RED);
        signupPanel.add(signupMsgLabel);

        signupBtn.addActionListener(e -> {
            String username = signupUsernameField.getText().trim();
            String password = new String(signupPasswordField.getPassword());

            // Validation: fields empty nahi honay chahiye
            if (username.isEmpty() || password.isEmpty()) {
                signupMsgLabel.setText("Username and password cannot be empty.");
                return;
            }
            // Username pehle se maujood hai?
            if (userDatabase.containsKey(username)) {
                signupMsgLabel.setText("Username already exists.");
                return;
            }

            userDatabase.put(username, password);  // Naya user add karo
            signupMsgLabel.setForeground(Color.GREEN);
            signupMsgLabel.setText("Signup successful! Please login.");
            // Fields clear karo
            signupUsernameField.setText("");
```



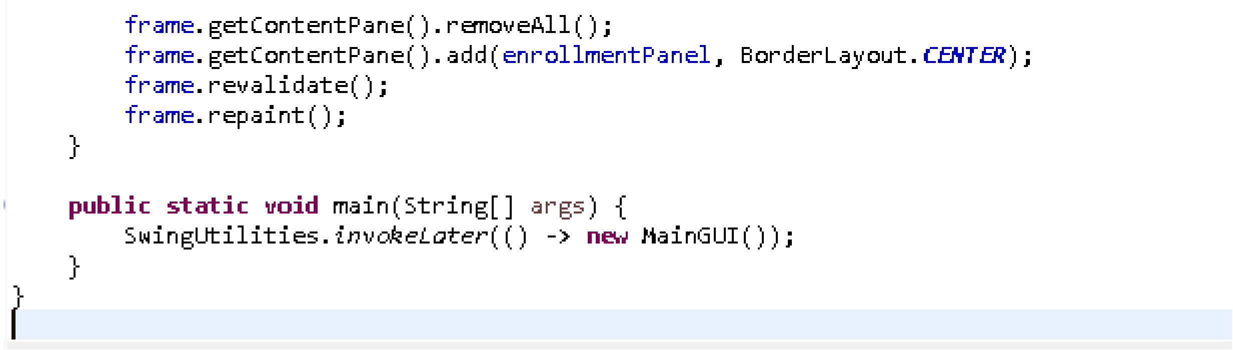
```
            // Signup ke baad login panel dikhao
            Timer timer = new Timer(1500, ev -> showLoginPanel());
            timer.setRepeats(false);
            timer.start();
        });

        frame.getContentPane().removeAll();
        frame.getContentPane().add(signupPanel, BorderLayout.CENTER);
        frame.revalidate();
        frame.repaint();
    }

    private void showLoginPanel() {
        loginPanel = new JPanel(new GridLayout(4, 2, 10, 10));
        loginPanel.setBorder(BorderFactory.createEmptyBorder(50, 100, 50, 100));

        loginPanel.add(new JLabel("Username:"));
        loginUsernameField = new JTextField();
        loginPanel.add(loginUsernameField);

        loginPanel.add(new JLabel("Password:"));
        loginPasswordField = new JPasswordField();
        loginPanel.add(loginPasswordField);

        loginBtn = new JButton("Login");
        loginPanel.add(loginBtn);

        loginMsgLabel = new JLabel("");
        loginMsgLabel.setForeground(Color.RED);
        loginPanel.add(loginMsgLabel);

        loginBtn.addActionListener(e -> {
            String username = loginUsernameField.getText().trim();
            String password = new String(loginPasswordField.getPassword());

            // User authentication check karo
            if (userDatabase.containsKey(username) && userDatabase.get(username).equals(password)) {
                showEnrollmentPanel();
            } else {
```



```java
        frame.getContentPane().removeAll();
        frame.getContentPane().add(loginPanel, BorderLayout.CENTER);
        frame.revalidate();
        frame.repaint();
    }

    private void showEnrollmentPanel() {
        enrollmentPanel = new JPanel(new BorderLayout());

        JPanel formPanel = new JPanel(new GridLayout(6, 2, 10, 10));
        formPanel.setBorder(BorderFactory.createEmptyBorder(20, 50, 20, 50));

        formPanel.add(new JLabel("Student Name:"));
        studentNameField = new JTextField();
        formPanel.add(studentNameField);

        formPanel.add(new JLabel("Father's Name:"));
        fatherNameField = new JTextField();
        formPanel.add(fatherNameField);

        formPanel.add(new JLabel("Semester:"));
        semesterField = new JTextField();
        formPanel.add(semesterField);

        formPanel.add(new JLabel("Program / Department:"));
        programField = new JTextField();
        formPanel.add(programField);

        formPanel.add(new JLabel("Select Course:"));
        courseDropdown = new JComboBox<>();
        for (Course c : FakeDatabase.courseList) {
            courseDropdown.addItem(c.getId() + " - " + c.getTitle());
        }
        formPanel.add(courseDropdown);

        enrollBtn = new JButton("Enroll");
        formPanel.add(enrollBtn);
```

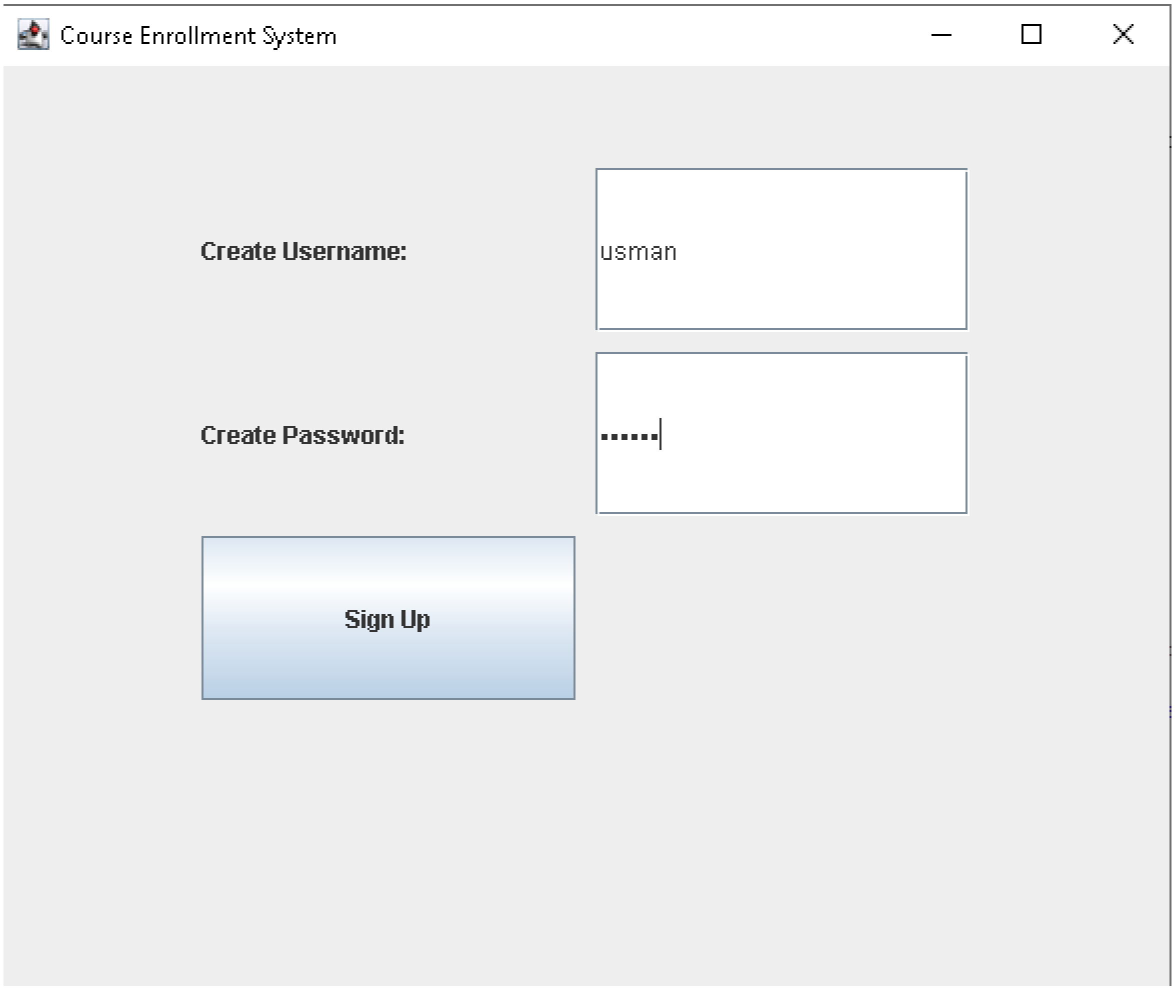


```
        enrollmentStatusArea = new JTextArea(10, 40);
        enrollmentStatusArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(enrollmentStatusArea);
        enrollmentPanel.add(scrollPane, BorderLayout.CENTER);

        enrollBtn.addActionListener(e -> {
            String name = studentNameField.getText().trim();
            String father = fatherNameField.getText().trim();
            String semester = semesterField.getText().trim();
            String program = programField.getText().trim();
            int courseIndex = courseDropdown.getSelectedIndex();

            // Check karo ke sab fields fill hain
            if (name.isEmpty() || father.isEmpty() || semester.isEmpty() || program.isEmpty()) {
                enrollmentStatusArea.append("× Please fill all fields!\n");
                return;
            }

            Course selectedCourse = FakeDatabase.courseList.get(courseIndex);
            Student student = new Student(
                "S" + (FakeDatabase.studentList.size() + 1),
                name,
                program,
                father,
                semester
            );
            FakeDatabase.studentList.add(student);

            boolean enrolled = enrollmentService.enroll(student, selectedCourse);
            if (enrolled) {
                enrollmentStatusArea.append("✅ " + name + " enrolled in " + selectedCourse.getTitle() + "\n");
            } else {
                enrollmentStatusArea.append("× Enrollment failed. No seats available in " + selectedCourse.getTitle() + "\n");
            }
        });
```



```
        frame.getContentPane().removeAll();
        frame.getContentPane().add(enrollmentPanel, BorderLayout.CENTER);
        frame.revalidate();
        frame.repaint();
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new MainGUI());
    }
}
```

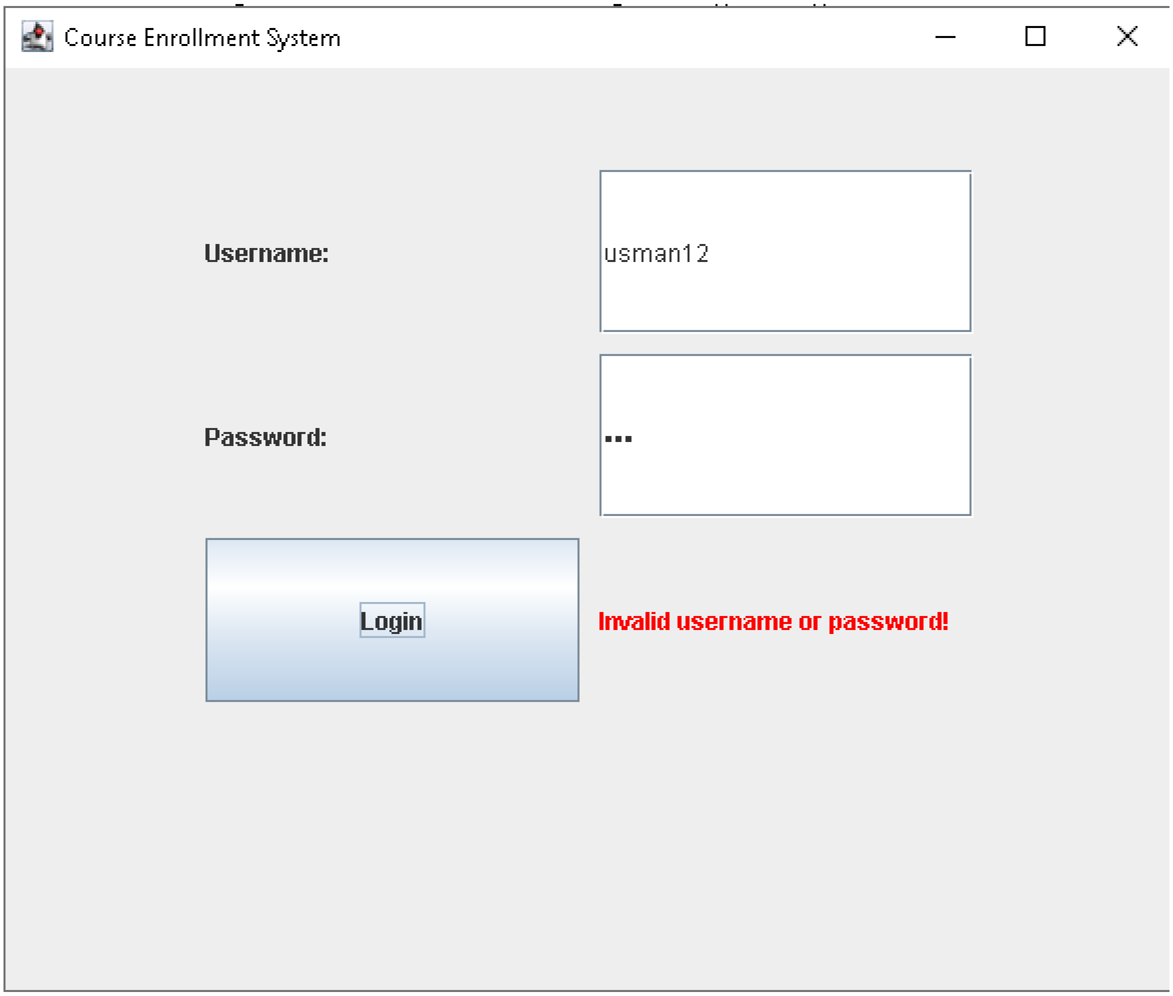
Output:



Course Enrollment System

Create Username: usman

Create Password: ••••••

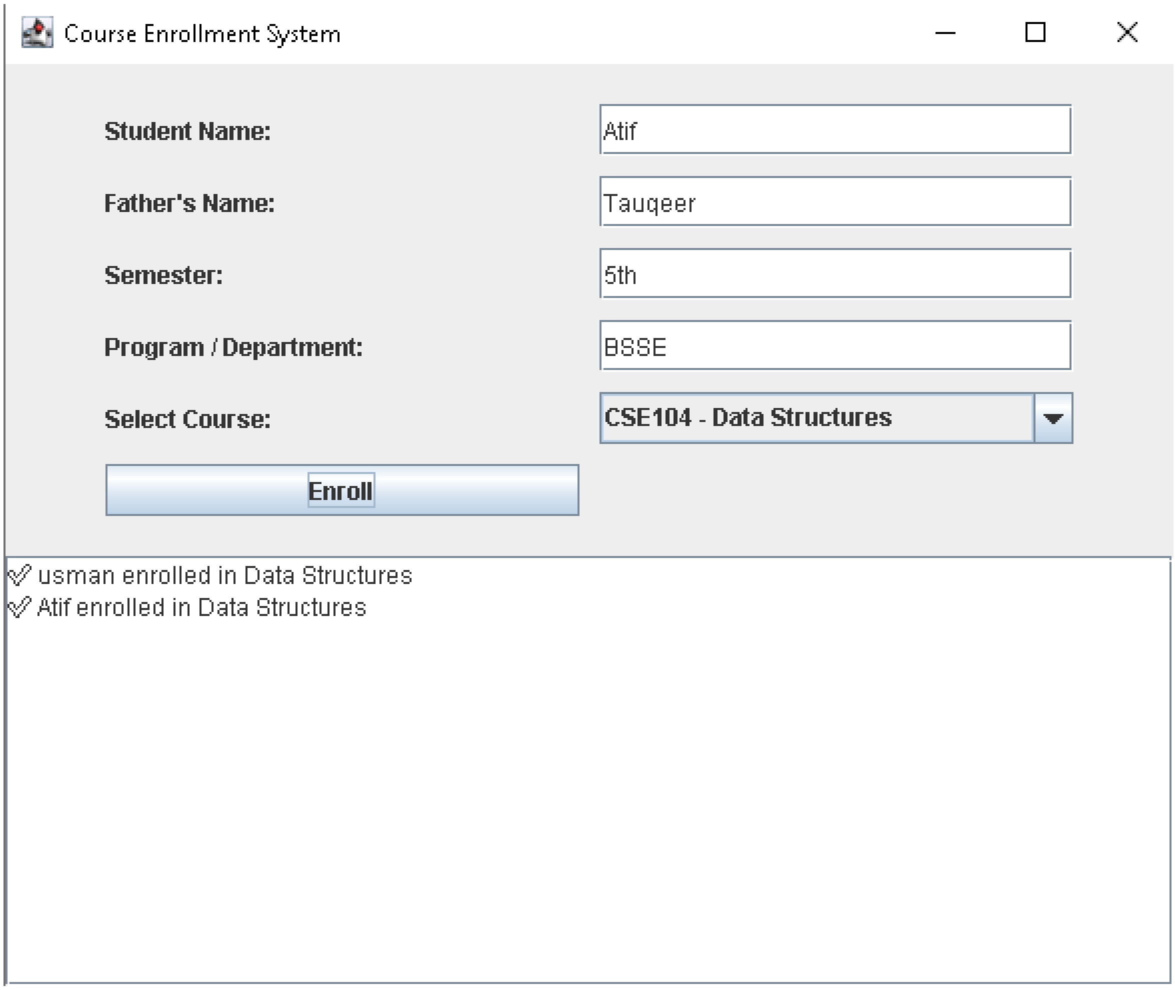


Course Enrollment System

Username: usman12

Password: •••

Invalid username or password!



Student Name: Atif

Father's Name: Tauqeer

Semester: 5th

Program / Department: BSSE

Select Course: CSE104 - Data Structures

```
✓ usman enrolled in Data Structures
✓ Atif enrolled in Data Structures
```