

JAVA MINI PROJECT REPORT

**Made By:- Group 2,
Dev Baliga
16010423073 & 16010423076
SY-(IT)-B3**

Project Theme: Student Admission Management System.

Problem Statement: Student Admission Management System addressed the challenges faced in the traditional student admission process, which often lacked efficiency and user-friendliness. The existing system did not provide a clear role-based access for users, making it difficult for students to submit their applications and for admins to manage and track applications effectively. We fixed these issues by developing a GUI that allows users to select their roles—either as a student or an admin—streamlining the admission process. Our solution included an enrollment form with validation checks to ensure accurate data submission, an admin view for easy management of student records, and a feature for students to check their application status.

Algorithm:

Step 1: Role Selection Page

- **1.1** Display a role selection window with three buttons:
 - "Student" to open the student enrollment form.
 - "Admin" to open the admin view.
 - "Check Application Status" to open the status check page.
- **1.2** On selecting "Student":
 - Close the role selection window.
 - Open the student enrollment form.
- **1.3** On selecting "Admin":
 - Close the role selection window.
 - Open the admin view.
- **1.4** On selecting "Check Application Status":
 - Close the role selection window.

- Open the check status page.

Step 2: Student Enrollment Form

- **2.1** Display the enrollment form with the following fields:
 - **Name**
 - **Personal Email**
 - **Marks** (Validate input: must be between 0–100).
 - **Age**
 - **Course** (Dropdown with course options).
- **2.2** Add two buttons:
 - **Submit** to save data to the database.
 - **Back** to return to the role selection page.
- **2.3** On pressing **Submit**:
 - Validate all input fields. If any are empty, show an error.
 - Validate marks to ensure they are within the 0–100 range.
 - If validation passes, save the data to the students table in the database.
 - Show a confirmation message and reset the form fields.
- **2.4** On pressing **Back**:
 - Close the form and reopen the role selection page.

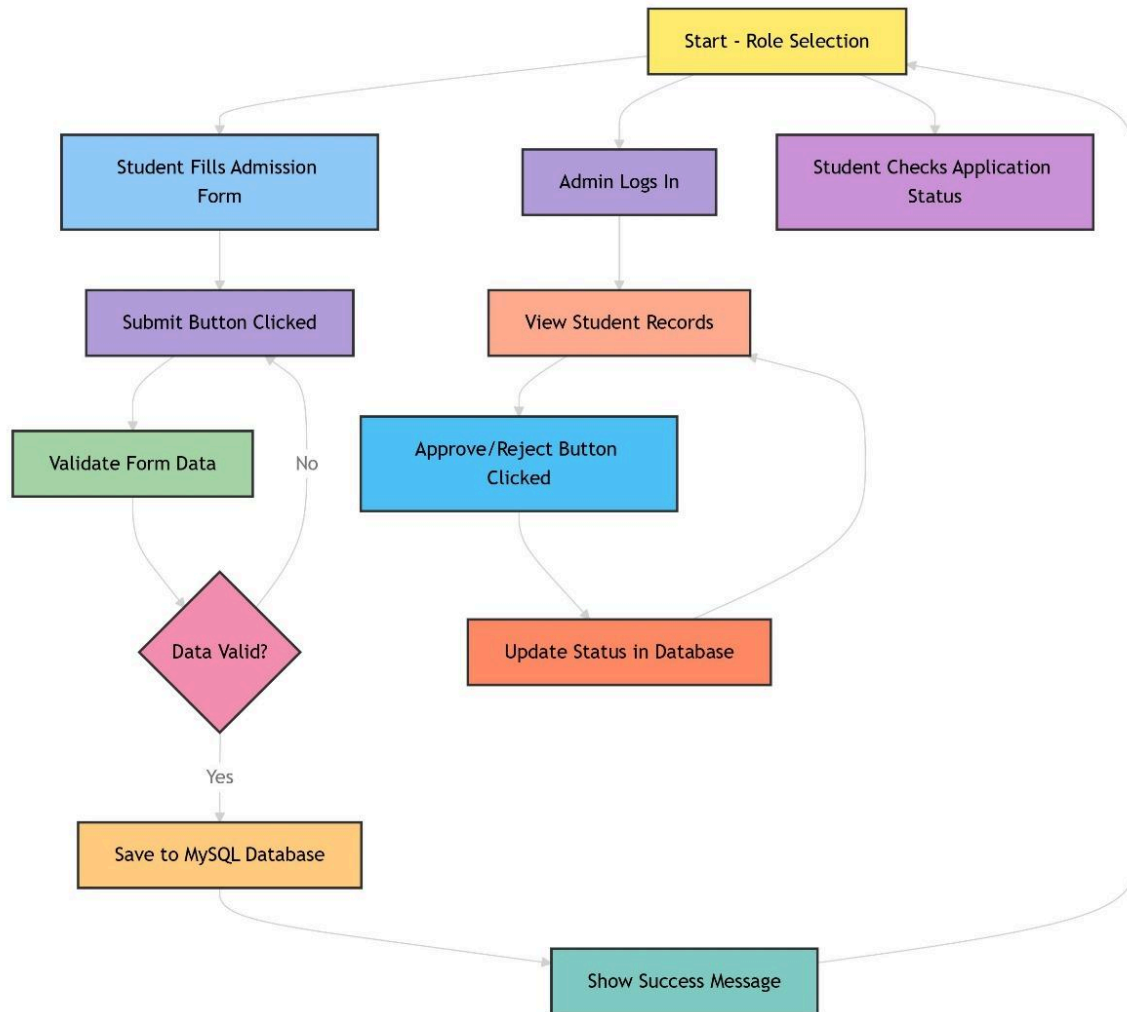
Step 3: Admin View Page

- **3.1** Display a table showing the following details of students from the students table:
 - ID, Name, Email, Age, Course, Status, and Action.
- **3.2** For each student record:
 - Display "Approve" and "Reject" buttons if the status is "Pending."
 - If the status is already "Approved" or "Rejected," display the status only.
- **3.3** On pressing the **Approve** button:
 - Update the status in the students table to "Approved."
 - Refresh the table to show the updated status.
- **3.4** On pressing the **Reject** button:
 - Update the status in the students table to "Rejected."
 - Refresh the table to show the updated status.

Step 4: Check Application Status Page

- **4.1** Display a form asking for the student's email.
- **4.2** Add two buttons:
 - **Check Status** to retrieve and display the application status.
 - **Back** to return to the role selection page.
- **4.3** On pressing **Check Status**:
 - Retrieve the status of the student from the students table using the entered email.
 - Display one of the following messages based on the retrieved status:
 - "Approved" message if the status is "Approved."
 - "Rejected" message if the status is "Rejected."
 - "Pending" message if the status is "Pending."
 - "No application found" if no record is found for the given email.
- **4.4** On pressing **Back**:
 - Close the check status page and reopen the role selection page.

Flowchart:



Code:-

AdminView.java Page:-

```

import javax.swing.*.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*.*;
import java.util.Vector;

```

```

public class AdminView extends JFrame {

```

```

    private JTable studentTable;
    private DefaultTableModel tableModel;

```

```

public AdminView() {
    setTitle("Admin View - Enrolled Students");
    setSize(600, 400);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());

    JLabel welcomeLabel = new JLabel("Welcome Admin!", SwingConstants.CENTER);
    welcomeLabel.setFont(new Font("Arial", Font.BOLD, 18));
    welcomeLabel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
    add(welcomeLabel, BorderLayout.NORTH);

    tableModel = new DefaultTableModel(new String[]{"ID", "Name", "Email", "Age", "Course",
"Status", "Action"}, 0);
    studentTable = new JTable(tableModel);
    studentTable.setFont(new Font("Arial", Font.PLAIN, 14));
    studentTable.setRowHeight(25);
    JScrollPane tableScrollPane = new JScrollPane(studentTable);
    add(tableScrollPane, BorderLayout.CENTER);

    viewStudents();
}

private void viewStudents() {
    String url = "jdbc:mysql://localhost:3306/college_system";
    String user = "root";
    String password = "";

    try (Connection con = DriverManager.getConnection(url, user, password)) {
        String query = "SELECT * FROM students";
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);

        tableModel.setRowCount(0);

        while (rs.next()) {
            Vector<Object> row = new Vector<>();
            row.add(rs.getInt("id"));
            row.add(rs.getString("name"));
            row.add(rs.getString("email"));
            row.add(rs.getInt("age"));
            row.add(rs.getString("course"));
            String status = rs.getString("status");
            row.add(status != null ? status : "Pending");
        }
    }
}

```

```

JPanel actionPanel = new JPanel();
JButton approveButton = new JButton("Approve");
JButton rejectButton = new JButton("Reject");

approveButton.addActionListener(e -> {
    try {
        updateStatus(rs.getInt("id"), "Approved");
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Error updating status: " +
ex.getMessage());
    }
});
rejectButton.addActionListener(e -> {
    try {
        updateStatus(rs.getInt("id"), "Rejected");
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Error updating status: " +
ex.getMessage());
    }
});

if ("Pending".equals(status)) {
    actionPanel.add(approveButton);
    actionPanel.add(rejectButton);
} else {
    actionPanel.add(new JLabel(status));
}

row.add(actionPanel);
tableModel.addRow(row);
}

} catch (SQLException ex) {
    JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());
}
}

private void updateStatus(int id, String newStatus) throws SQLException {
    String url = "jdbc:mysql://localhost:3306/college_system";
    String user = "root";
    String password = "";

    try (Connection con = DriverManager.getConnection(url, user, password)) {

```

```

        String updateQuery = "UPDATE students SET status = ? WHERE id = ?";
        PreparedStatement stmt = con.prepareStatement(updateQuery);
        stmt.setString(1, newStatus);
        stmt.setInt(2, id);
        stmt.executeUpdate();

        JOptionPane.showMessageDialog(this, "Status updated to: " + newStatus);
        viewStudents();
    }
}

public static void main(String[] args) {
    new AdminView().setVisible(true);
}
}

```

Form.java Page:-

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class Form extends JFrame {

    private JTextField nameField, emailField, ageField, marksField;
    private JComboBox<String> courseDropdown;
    private JButton submitButton, backButton;

    public Form(String role) {
        // Create the form frame
        setTitle(role + " Enrollment Form");
        setSize(500, 350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridBagLayout()); // Using GridBagLayout for more control
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(5, 5, 5, 5); // Padding for spacing between components

        // Welcome message at the top
        JLabel welcomeLabel = new JLabel("Welcome Student, please fill the below form for admission.");
    }
}

```

```

welcomeLabel.setFont(new Font("Arial", Font.BOLD, 16));
welcomeLabel.setHorizontalAlignment(SwingConstants.CENTER);
welcomeLabel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
gbc.gridwidth = 2; // Span across two columns
gbc.gridx = 0;
gbc.gridy = 0;
add(welcomeLabel, gbc);

// Name field
JLabel nameLabel = new JLabel("Name:");
nameLabel.setFont(new Font("Arial", Font.PLAIN, 14));
nameField = new JTextField(20);
nameLabel.setLabelFor(nameField); // Associate label with field

gbc.gridwidth = 1; // Reset gridwidth to 1 for next fields
gbc.gridx = 0;
gbc.gridy = 1;
add(nameLabel, gbc);

gbc.gridx = 1;
gbc.gridy = 1;
nameField.setPreferredSize(new Dimension(200, 30)); // Smaller input field size
add(nameField, gbc);

// Somaiya Email field
JLabel emailLabel = new JLabel("Personal Email:");
emailLabel.setFont(new Font("Arial", Font.PLAIN, 14));
emailField = new JTextField(20);
emailLabel.setLabelFor(emailField); // Associate label with field

gbc.gridx = 0;
gbc.gridy = 2;
add(emailLabel, gbc);

gbc.gridx = 1;
gbc.gridy = 2;
emailField.setPreferredSize(new Dimension(200, 30));
add(emailField, gbc);

// Marks field (NEW FIELD)
JLabel marksLabel = new JLabel("Marks (0-100):");
marksLabel.setFont(new Font("Arial", Font.PLAIN, 14));
marksField = new JTextField(5);
marksLabel.setLabelFor(marksField); // Associate label with field

```



```
gbc.gridx = 0;
gbc.gridy = 3;
add(marksLabel, gbc);
```

```
gbc.gridx = 1;
gbc.gridy = 3;
marksField.setPreferredSize(new Dimension(100, 30));
add(marksField, gbc);
```

```
// Age field
JLabel ageLabel = new JLabel("Age:");
ageLabel.setFont(new Font("Arial", Font.PLAIN, 14));
ageField = new JTextField(5);
ageLabel.setLabelFor(ageField); // Associate label with field
```

```
gbc.gridx = 0;
gbc.gridy = 4;
add(ageLabel, gbc);
```

```
gbc.gridx = 1;
gbc.gridy = 4;
ageField.setPreferredSize(new Dimension(100, 30)); // Smaller input box
add(ageField, gbc);
```

```
// Course dropdown
JLabel courseLabel = new JLabel("Course:");
courseLabel.setFont(new Font("Arial", Font.PLAIN, 14));
String[] courses = {
    "Computer Engineering(Comps)",
    "Information Technology Engineering(IT)",
    "Artificial Intelligence and Data Science(AI&DS)",
    "Computer & Communication Engineering(CCE)",
    "Electronics and Telecommunication Engineering(EXTC)"
};
courseDropdown = new JComboBox<>(courses);
courseLabel.setLabelFor(courseDropdown); // Associate label with field
```

```
gbc.gridx = 0;
gbc.gridy = 5;
add(courseLabel, gbc);
```

```
gbc.gridx = 1;
gbc.gridy = 5;
```

```

courseDropdown.setPreferredSize(new Dimension(300, 30)); // Increased dropdown width
add(courseDropdown, gbc);

// Submit button
submitButton = new JButton("Submit");
submitButton.setFont(new Font("Arial", Font.PLAIN, 14));
gbc.gridwidth = 2; // Make the button span across two columns
gbc.gridx = 0;
gbc.gridy = 6;
gbc.insets = new Insets(20, 5, 5, 5); // Add some extra padding before the button
add(submitButton, gbc);

// Back button
backButton = new JButton("Back");
backButton.setFont(new Font("Arial", Font.PLAIN, 14));
gbc.gridx = 0;
gbc.gridy = 7;
gbc.insets = new Insets(10, 5, 10, 5); // Padding for the back button
add(backButton, gbc);

// Action for submit button
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        String email = emailField.getText();
        String age = ageField.getText();
        String marks = marksField.getText();
        String course = courseDropdown.getSelectedItem().toString();

        // Validate marks field (should be between 0 and 100)
        try {
            int marksValue = Integer.parseInt(marks);
            if (marksValue < 0 || marksValue > 100) {
                throw new NumberFormatException();
            }
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(Form.this, "Please enter marks between 0 and
100.", "Invalid Marks", JOptionPane.ERROR_MESSAGE);
            return; // Stop form submission if validation fails
        }

        if (name.isEmpty() || email.isEmpty() || age.isEmpty() || marks.isEmpty()) {
            // Display an error if fields are empty

```

```

        JOptionPane.showMessageDialog(Form.this, "Please fill out all fields!", "Error",
JOptionPane.ERROR_MESSAGE);
    } else {
        saveData(name, email, age, marks, course); // Save student data to database
        // Show confirmation message after successful submission
        JOptionPane.showMessageDialog(Form.this, "Form has been submitted
successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
        // Optionally, you can clear the form after submission
        nameField.setText("");
        emailField.setText("");
        ageField.setText("");
        marksField.setText("");
        courseDropdown.setSelectedIndex(0);
    }
}
});

```

```

// Action for back button
backButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Close current window and return to RoleSelection page
        dispose(); // Close the form window
        new RoleSelectionPage().setVisible(true); // Open the RoleSelection page
    }
});
}

```

```

// Method to save form data to MySQL
private void saveData(String name, String email, String age, String marks, String course) {
    try {
        // Database connection details
        String url = "jdbc:mysql://localhost:3306/college_system";
        String username = "root"; // Change this to your database username
        String password = "root"; // Change this to your database password

        // Create a connection to the database
        Connection conn = DriverManager.getConnection(url, username, password);

        // SQL query to insert data into the student table
        String query = "INSERT INTO students (name, email, age, marks, course) VALUES (?,
?, ?, ?, ?)";

        // Create a PreparedStatement
        PreparedStatement stmt = conn.prepareStatement(query);
    }
}

```

```

        stmt.setString(1, name);
        stmt.setString(2, email);
        stmt.setString(3, age);
        stmt.setString(4, marks);
        stmt.setString(5, course);

        // Execute the insert operation
        int rowsAffected = stmt.executeUpdate();
        if (rowsAffected > 0) {
            System.out.println("Data saved successfully!");
        } else {
            System.out.println("Failed to save data.");
        }

        // Close the connection
        stmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(Form.this, "Error saving data to database!", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

public static void main(String[] args) {
    new Form("Student").setVisible(true);
}
}

```

Role Selection Page.java :-

```

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

public class RoleSelectionPage extends JFrame {

    public RoleSelectionPage() {
        // Setup the frame
        setTitle("Select Role");
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
    }
}

```

```

// Add logo image at the top center (make sure to put the logo image in the project folder)
ImageIcon logolcon = new ImageIcon("logo.png"); // Logo image file
Image logolImage = logolcon.getImage().getScaledInstance(100, 100,
Image.SCALE_SMOOTH); // Resize the image
logolcon = new ImageIcon(logolImage); // Update the ImageIcon with the resized image
JLabel logoLabel = new JLabel(logolcon);
logoLabel.setHorizontalAlignment(SwingConstants.CENTER);
add(logoLabel, BorderLayout.NORTH);

// Create a panel for "Select your role" and buttons
JPanel panel = new JPanel();
panel.setLayout(new GridLayout(4, 1, 10, 10)); // Grid layout for better centering and
spacing

// "Select your role" label
JLabel label = new JLabel("Select your role");
JLabel label2 = new JLabel("Welcome to Somaiya Vidyavihar Admission Portal ");
label.setFont(new Font("Arial", Font.BOLD, 18));
label.setHorizontalAlignment(SwingConstants.CENTER);
label2.setFont(new Font("Arial", Font.BOLD, 18));
label2.setHorizontalAlignment(SwingConstants.CENTER);
panel.add(label2);
panel.add(label);

// Role selection buttons
JPanel buttonPanel = new JPanel(new FlowLayout());
JButton studentButton = new JButton("Student");
JButton adminButton = new JButton("Admin");
JButton statusCheckButton = new JButton("Check Application Status");

// Customize the buttons
studentButton.setFont(new Font("Arial", Font.PLAIN, 14));
adminButton.setFont(new Font("Arial", Font.PLAIN, 14));
statusCheckButton.setFont(new Font("Arial", Font.PLAIN, 14));
studentButton.setPreferredSize(new Dimension(120, 40));
adminButton.setPreferredSize(new Dimension(120, 40));
statusCheckButton.setPreferredSize(new Dimension(180, 40));

// Add buttons to buttonPanel
buttonPanel.add(studentButton);
buttonPanel.add(adminButton);
buttonPanel.add(statusCheckButton);

```

```

// Add buttonPanel to the main panel
panel.add(buttonPanel);

// Button action listeners
studentButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new Form("Student").setVisible(true); // Show Student Form
        dispose(); // Close the role selection window
    }
});

adminButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new AdminView().setVisible(true); // Show Admin View
        dispose(); // Close the role selection window
    }
});

// Action listener for status check button
statusCheckButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new CheckStatusPage().setVisible(true); // Show Check Status Page
        dispose(); // Close the role selection window
    }
});

// Add the panel to the center of the frame
add(panel, BorderLayout.CENTER);
}

public static void main(String[] args) {
    // Show role selection page
    new RoleSelectionPage().setVisible(true);
}
}

```

CheckStatusPage.java :-

```

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

public class CheckStatusPage extends JFrame {

```

```

private JTextField emailField;
private JButton checkButton, backButton;
private JLabel resultLabel;

public CheckStatusPage() {
    setTitle("Check Application Status");
    setSize(400, 250);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new GridBagLayout());

    GridBagConstraints gbc = new GridBagConstraints();
    gbc.insets = new Insets(5, 5, 5, 5);

    JLabel emailLabel = new JLabel("Enter Your Email:");
    emailField = new JTextField(20);
    gbc.gridx = 0;
    gbc.gridy = 0;
    add(emailLabel, gbc);

    gbc.gridx = 1;
    add(emailField, gbc);

    checkButton = new JButton("Check Status");
    gbc.gridx = 0;
    gbc.gridy = 1;
    gbc.gridwidth = 2;
    add(checkButton, gbc);

    resultLabel = new JLabel("");
    gbc.gridy = 2;
    add(resultLabel, gbc);

    backButton = new JButton("Back");
    gbc.gridy = 3;
    add(backButton, gbc);

    checkButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String email = emailField.getText().trim();
            if (email.isEmpty()) {
                JOptionPane.showMessageDialog(CheckStatusPage.this, "Please enter an email!",
"Error", JOptionPane.ERROR_MESSAGE);
            } else {

```

```

        checkApplicationStatus(email);
    }
}
});

backButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
        new RoleSelectionPage().setVisible(true);
    }
});
}

private void checkApplicationStatus(String email) {
    String url = "jdbc:mysql://localhost:3306/college_system";
    String username = "root";
    String password = "root";

    try (Connection conn = DriverManager.getConnection(url, username, password)) {
        String query = "SELECT status FROM students WHERE email = ?";
        PreparedStatement stmt = conn.prepareStatement(query);
        stmt.setString(1, email);
        ResultSet rs = stmt.executeQuery();

        if (rs.next()) {
            String status = rs.getString("status");
            if ("Approved".equals(status)) {
                resultLabel.setText("Congratulations! Your application has been approved. Please visit the Somaiya Vidyavihar University to confirm your admission.");
            } else if ("Rejected".equals(status)) {
                resultLabel.setText("We regret to inform you that your application was reviewed but was not found suitable. Thank you for your interest in Somaiya Vidyavihar University.");
            } else {
                resultLabel.setText("Application Status: Pending");
            }
        } else {
            resultLabel.setText("No application found for the given email.");
        }
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(CheckStatusPage.this, "Error fetching application status!", "Error", JOptionPane.ERROR_MESSAGE);
    }
}


```



```
}  
  
public static void main(String[] args) {  
    new CheckStatusPage().setVisible(true);  
}  
}
```

Output:

Select Role



Welcome to Somaiya Vidyavihar Admission Portal

Select your role

Student

Admin

Check Application St...

meet.google.com is sharing your screen. Stop sharing Hide

Student Enrollment Form

Welcome Student, please fill the below form for admission.

Name:

Personal Email:

Marks (0-100):

Age:

Course:

Submit

Back

Admin View - Enrolled Students

Welcome Admin!

ID	Name	Email	Age	Course	Marks	Actions
1	Dev	devbaliga@gmail.com	19	Computer Engineering(Comps)	100	<div>AcceptReject</div>
2	Ritesh	ritesh@gmail.com	19	Information Technology Engine...	77	<div>AcceptReject</div>
3	Dev	devbaliga@gmail.com	19	Computer Engineering(Comps)	98	<div>AcceptReject</div>
4	Omkar Dabade	ogdabde@gmail.com	19	Artificial Intelligence and Data S...	11	<div>AcceptReject</div>
5	Ritesh	riteshha@gmail.com	18	Computer Engineering(Comps)	98	<div>AcceptReject</div>

Back

29°

Search

ENG

IN

20:51

03-11-2024

Check Application Status

Enter Your Email:

Check Status

Back

29°

Search

ENG

IN

23:32

05-11-2024

Backend MySql code:

```
DROP database college_system;
```

```
CREATE DATABASE IF NOT EXISTS college_system;
```

```
USE college_system;
```

```
-- Table for storing student data
```

```
CREATE TABLE IF NOT EXISTS students (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    age INT NOT NULL,  
    marks INT NOT NULL CHECK (marks >= 0 AND marks <= 100), -- Added marks field  
    with validation (0-100)  
    course VARCHAR(255) NOT NULL,  
    status ENUM('Pending', 'Accepted', 'Rejected') DEFAULT 'Pending' -- Added status field  
    with default 'Pending'  
);
```

```
-- Table for storing user roles
```

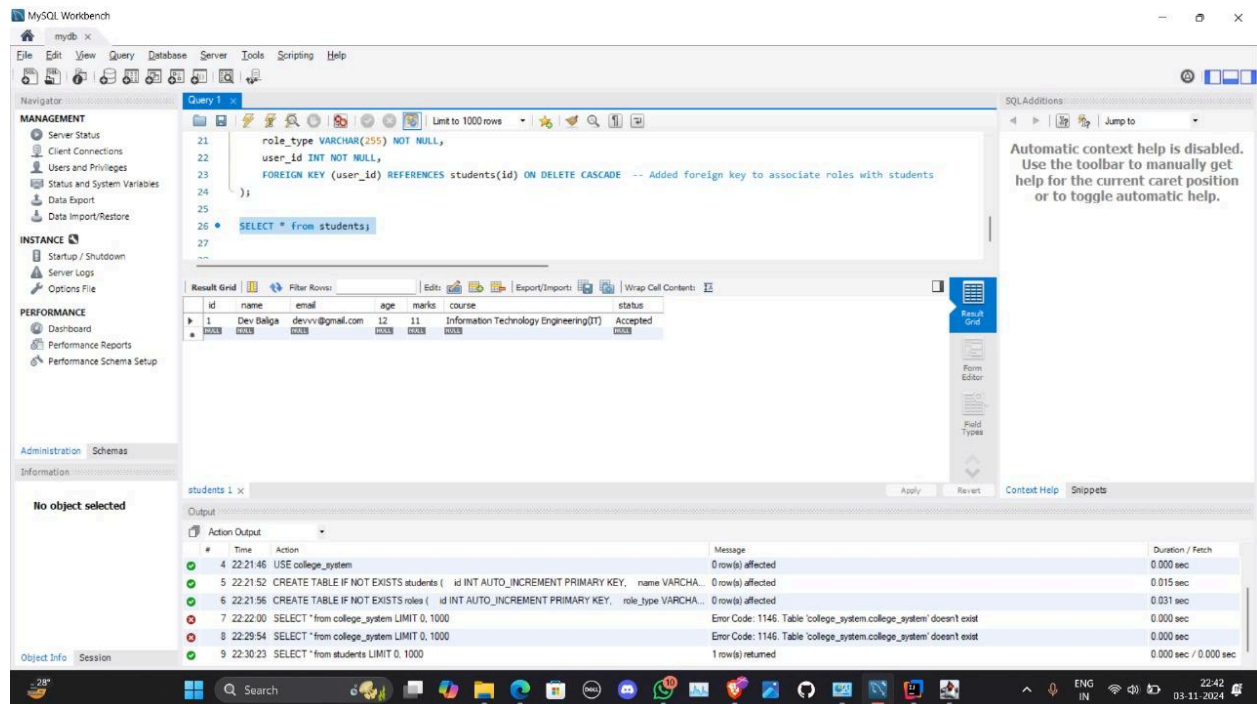
```
CREATE TABLE IF NOT EXISTS roles (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    role_type VARCHAR(255) NOT NULL,  
    user_id INT NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES students(id) ON DELETE CASCADE -- Added  
    foreign key to associate roles with students  
);
```

```
CREATE TABLE applications (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    age INT NOT NULL,  
    course VARCHAR(50) NOT NULL,  
    marks FLOAT NOT NULL,  
    status ENUM('Under Review', 'Accepted', 'Rejected') DEFAULT 'Under Review'  
);
```

```
DROP TABLE applications;
```

```
SELECT * from students;
```

Output:



Conclusion :

In creating the Student Admission Management System, I learned how to design a user-friendly interface using Swing in Java, which allows different user roles to navigate the application easily. I gained practical experience in implementing input validation to ensure data integrity when students submit their information. Additionally, I understood how to manage data in a MySQL database using JDBC, which helped me connect the enrollment form to the students table and see how backend processes support frontend functionalities. Working in IntelliJ also improved my coding efficiency. Overall, this mini-project enhanced my programming skills and deepened my understanding of how various components, including frontend, backend, and database interactions, work together in a real-world application.