

**Project Report
On**

STUDENT BIO DATA MANAGEMENT SYSTEM

During

II Year I Semester Summer

Submitted to

The Department of CSE-Cyber Security

In partial fulfillment of the academic requirements of

Jawaharlal Nehru Technological University

For

The award of the degree of

Bachelor of Technology

In

CSE-Cyber Security

By

| | |
|------------------------------|-------------------|
| Thatipamula Rahul | 22311A6214 |
| Kodyaram Sathwika | 22311A6208 |
| Yaswanth Valurouthu | 22311A6213 |
| Pittala Pravallika | 22311A6222 |
| Porandla Mahendranath | 22311A6229 |
| Aduri Ashish | 22311A6261 |
| Keesara Sri Ranjitha | 22311A6257 |
| Kaveti Rahul | 23315A6208 |

K. Gnaneshwar

Asst. Prof.

Dept. of CSE- Cyber Security



CONTENTS

| S. NO | TOPIC NAME | PAGE NO. |
|--------------|----------------------|-----------------|
| 1. | Abstract | 1 |
| 2. | Introduction | 2 |
| 3. | Code | 3 - 31 |
| 4. | Output(Screen Shots) | 32-34 |
| 5. | Conclusion | 35 |

Abstract

The Student BioData Management System is a straightforward software tool designed to assist educational institutions in organizing and managing student information efficiently. The system provides basic functionalities for storing, retrieving, and updating student records.

Key features of the system include:

Basic Data Management: Users can input and store essential student details such as name, ID number, contact information, and course enrollment.

User-friendly Interface: The system offers a simple and intuitive interface, making it easy for users to navigate and perform basic operations without extensive training.

Data Accessibility: Administrators can quickly retrieve student records when needed, facilitating smooth administrative processes within the institution.

Minimalist Design: With a focus on simplicity, the system eliminates unnecessary complexities, ensuring ease of use for administrators with varying levels of technical expertise.

The Student BioData Management System provides a hassle-free solution for educational institutions seeking a straightforward tool for managing student information effectively.

Introduction

In today's educational landscape, efficient management of student information is crucial for the smooth functioning of educational institutions. The Student BioData Management System offers a solution to this challenge by providing a streamlined approach to organizing and accessing student records.

With the increasing volume of student data generated in educational institutions, traditional paper-based methods of record-keeping are no longer sufficient. Manual processes are prone to errors, time-consuming, and often result in inefficiencies in administrative tasks.

The Student BioData Management System aims to address these issues by digitizing the process of managing student information. By leveraging technology, the system simplifies data entry, retrieval, and maintenance, allowing administrators to focus more on their core responsibilities.

This introduction sets the stage for understanding the significance of the Student BioData Management System in modern educational environments. In the subsequent sections, we will delve deeper into the features, functionalities, and benefits of the system, highlighting its potential to streamline administrative processes and enhance overall efficiency within educational institutions.

Code

LoginWindow.java

```
package com.mycompany.studentbiodata;

import javax.swing.JOptionPane;

public class LoginWindow extends javax.swing.JFrame {

    public LoginWindow() {
        initComponents();
    }

    private void loginBtnActionPerformed(java.awt.event.ActionEvent evt) {

        String userName = userNameField.getText();

        String password = passwordField.getText();

        String query = "select * from admindata where username =? and password = ? ";

        ProcessData processData = new ProcessData(query);

        if(userName.isEmpty() || password.isEmpty()){

            System.out.println(userName + "Hello" +password);

            statusMessage.setVisible(true);

            if(userName.isEmpty() && password.isEmpty())

                statusMessage.setText("Please enter userName and Password");

            else if(password.isEmpty())

                statusMessage.setText("Please Enter password");

            else

                statusMessage.setText("Please enter username");

        }

        else if(processData.verifyAdminDetails(userName, password)){

            this.dispose();

            JOptionPane.showMessageDialog(null, "Login Successful");

            new AdminPanel().setVisible(true);

        }

        else{
```

```

        statusMessage.setText("No admin Details Found...Check your credentials");
    } }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new LoginWindow().setVisible(true);
            } });
    }

    private javax.swing.JLabel adminLabel;
    private javax.swing.JButton loginBtn;
    private javax.swing.JPasswordField passwordField;
    private javax.swing.JLabel passwordLabel;
    private javax.swing.JLabel statusMessage;
    private javax.swing.JTextField userNameField;
    private javax.swing.JLabel userNameLabel;
}

```

AddStudentPanel.java

```

package com.mycompany.studentbiodata;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.sql.*;
import java.util.ArrayList;
import java.util.Enumeraion;
import javax.swing.*;

public class AddStudentPanel extends javax.swing.JPanel {

    public AddStudentPanel() {

        setGetStudentData = new SetGetStudentData();

        initComponents();
    }
}

```

```

private void rollNumberFieldFocusLost(java.awt.event.FocusEvent evt) {
    String rollNumber = rollNumberField.getText();
    if (rollNumberCheck(rollNumber)) {
        JOptionPane.showMessageDialog(null, "Roll number already exists!");
        rollNumberField.setText("");
    }
}

public boolean rollNumberCheck(String rollNumber){
    try {
        Connection con = GetConnection.createConnection();
        String query = "Select * from studentBioData where rollnumber = ?";
        PreparedStatement statement = con.prepareStatement(query);
        statement.setString(1, rollNumberField.getText());
        ResultSet rs = statement.executeQuery();
        return rs.next();
    } catch (SQLException ex) {
        String errorMessage = ex.getMessage();
        JOptionPane.showMessageDialog(null, "An error occurred: " + errorMessage, "Error",
JOptionPane.ERROR_MESSAGE);}
    return false;}

private void photoSelectActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser fileChooser = new JFileChooser();
    File selectedFile =null;
    int returnValue = fileChooser.showOpenDialog(null);
    ArrayList<String> ALLOWED_EXTENSIONS = new ArrayList<String>();
    ALLOWED_EXTENSIONS.add(".jpg"); ALLOWED_EXTENSIONS.add(".jpeg");
    ALLOWED_EXTENSIONS.add(".png"); ALLOWED_EXTENSIONS.add(".gif");
    if (returnValue == JFileChooser.APPROVE_OPTION) {
        selectedFile = fileChooser.getSelectedFile();
        ProcessData pd = new ProcessData();
        String extension = pd.getFileExtension(selectedFile.getAbsolutePath());

```

```

try{
    if (!ALLOWED_EXTENSIONS.contains(extension.toLowerCase())) {
        throw new IllegalArgumentException("Photo Format Error");
    }
    else{
        try {
            photoSelect.setText(selectedFile.getPath());
            uploadPhotoToDatabase(selectedFile);
        } catch (SQLException | IOException ex) {
            String errorMessage = ex.getMessage();
            JOptionPane.showMessageDialog(null, "An error occurred: " + errorMessage, "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }} catch (IllegalArgumentException iae){
        selectedFile=null;
        setGetStudentData.setPhotoPath(null);
        photoSelect.setText("select Photo");
        JOptionPane.showMessageDialog(null, "Error:  " + iae.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }}}

SetGetStudentData setGetStudentData=null;

private void submitBtnActionPerformed(java.awt.event.ActionEvent evt) {
    setGetStudentData.setName(studentNameField.getText());
    setGetStudentData.setRollNumber(rollNumberField.getText());
    setGetStudentData.setFatherName(fatherNameField.getText());
    setGetStudentData.setMotherName(motherNameField.getText());
    setGetStudentData.setDateOfBirth(dateOfBirthField.getText());
    setGetStudentData.setEmail(emailIdField.getText());
    setGetStudentData.setBranch(branchField.getText());
    if(attendaceField.getText().isEmpty()){
        setGetStudentData.setAttendancePercentage(0.0);

```



```

    }
    else{

setGetStudentData.setAttendancePercentage(Double.parseDouble(attendaceField.getText()));

    }

    if(gpaField.getText().isEmpty()){          setGetStudentData.setCgpa(0.0); }
    else{

        setGetStudentData.setCgpa(Double.parseDouble(gpaField.getText()));
    }setGetStudentData.setAddress(addressField.getText());

    setGetStudentData.setCourseDuration((String) courseDurationField.getSelectedItem());

    Enumeration<AbstractButton> buttons = gender.getElements();

    while (buttons.hasMoreElements()) {

        AbstractButton button = buttons.nextElement();

        if (button.isSelected()) { setGetStudentData.setGender(button.getText()); break;

        } }

    String query = "INSERT INTO STUDENTBIODATA (NAME, FATHERNAME,
MOTHERNAME, DATEOFBIRTH, GENDER, EMAIL, BRANCH,
ATTENDANCEPERCENTAGE, CGPA, COURSEDURATION, ADDRESS,
PHOTO,ROLLNUMBER) "

        + "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

    ProcessData processData = new ProcessData(query);

    if (processData.nullCheck(setGetStudentData)) {

        boolean status = processData.addStudent(setGetStudentData);

        if (status) {

            JOptionPane.showMessageDialog(null, "Data successfully saved.");

        } else {

            JOptionPane.showMessageDialog(null, "Failed to save data ...Try again");

        }

    }

    else {

        JOptionPane.showMessageDialog(null, "Enter all Fields to Save");

    }

}

private void backBtnActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        adminPanelRef.adminHomePanel.remove(this);
adminPanelRef.adminHomePanel.add(adminPanelRef.homePanel);

        adminPanelRef.adminHomePanel.repaint();
adminPanelRef.adminHomePanel.revalidate();

    }

    AdminPanel adminPanelRef=null;

    public void AdminPanelReference(AdminPanel adminPanelRef){

        this.adminPanelRef = adminPanelRef;

    }

    private void emailIdFieldFocusLost(java.awt.event.FocusEvent evt) {

        EmailVerifier ev = new EmailVerifier(emailIdField.getText());

        if(ev.verifyEmail()){

        }else{

            JOptionPane.showMessageDialog(null, "Email Format is not Correct");

            emailIdField.setText(null);}}

    private void uploadPhotoToDatabase(File selectedFile) throws SQLException, IOException
    {

        try {

            FileInputStream studentPhotoFile = new FileInputStream(selectedFile);

            setGetStudentData.setPhotoPath(studentPhotoFile);

        } catch (Exception ex) {

            String errorMessage = ex.getMessage();

            JOptionPane.showMessageDialog(null, "An error occurred: " + errorMessage, "Error",
JOptionPane.ERROR_MESSAGE);

        }

    }

    private javax.swing.JTextArea addressField; private javax.swing.JTextField attendaceField;
private javax.swing.JButton backBtn; private javax.swing.JTextField branchField; private
javax.swing.JComboBox<String> courseDurationField; private javax.swing.JTextField
dateOfBirthField; private javax.swing.JTextField emailIdField; private
javax.swing.JTextField fatherNameField; private javax.swing.JRadioButton female; private
javax.swing.ButtonGroup gender; private javax.swing.JTextField gpaField; private
javax.swing.JLabel jLabel1; private javax.swing.JLabel jLabel10; private javax.swing.JLabel
jLabel11; private javax.swing.JLabel jLabel12; private javax.swing.JLabel jLabel13; private
javax.swing.JLabel jLabel15; private javax.swing.JLabel jLabel2; private javax.swing.JLabel
jLabel3; private javax.swing.JLabel jLabel4; private javax.swing.JLabel jLabel6; private

```

```

javax.swing.JLabel jLabel7; private javax.swing.JLabel jLabel8; private javax.swing.JLabel
jLabel9; private javax.swing.JScrollPane jScrollPane2; private javax.swing.JRadioButton
male; private javax.swing.JTextField motherNameField; private javax.swing.JRadioButton
others; private javax.swing.JButton photoSelect; private javax.swing.JTextField
rollNumberField; private javax.swing.JTextField studentNameField; private
javax.swing.JButton submitBtn;

```

```

}

```

AdminPanel.java

```

package com.mycompany.studentbiodata;

```

```

public class AdminPanel extends javax.swing.JFrame {

```

```

    public AdminPanel() {

```

```

        initComponents();

```

```

    }

```

```

    private void updateStudentDetailsActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        UpdateStudentPanel usp = new UpdateStudentPanel();

```

```

        usp.setVisible(true);    usp.AdminPanelReference(this);

```

```

        adminHomePanel.removeAll();    adminHomePanel.add(usp);

```

```

        adminHomePanel.repaint();    adminHomePanel.revalidate();

```

```

    }

```

```

    private void displayAllStudentsActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        DisplayAllStudentPanel dasp = new DisplayAllStudentPanel();

```

```

        dasp.setVisible(true);    dasp.AdminPanelReference(this);

```

```

        adminHomePanel.removeAll(); adminHomePanel.add(dasp);

```

```

        adminHomePanel.repaint();    adminHomePanel.revalidate();

```

```

    }

```

```

    private void checkStudentActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        CheckStudentPanel csp = new CheckStudentPanel();

```

```

        csp.setVisible(true);    csp.AdminPanelReference(this);

```

```

        adminHomePanel.removeAll();    adminHomePanel.add(csp);

```

```

        adminHomePanel.repaint();    adminHomePanel.revalidate();

```

```

    }

```

```

private void removeStudentActionPerformed(java.awt.event.ActionEvent evt) {
    RemoveStudentPanel rsp = new RemoveStudentPanel();
    rsp.setVisible(true);    rsp.AdminPanelReference(this);
    adminHomePanel.removeAll();    adminHomePanel.add(rsp);
    adminHomePanel.repaint();    adminHomePanel.revalidate();
}

private void addStudentActionPerformed(java.awt.event.ActionEvent evt) {
    AddStudentPanel asp = new AddStudentPanel();
    asp.setVisible(true);    asp.AdminPanelReference(this);
adminHomePanel.removeAll();

    adminHomePanel.add(asp); adminHomePanel.repaint(); adminHomePanel.revalidate();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new AdminPanel().setVisible(true);
        }
    });
}

private javax.swing.JButton addStudent;

public javax.swing.JPanel adminHomePanel;    private    javax.swing.JButton
checkStudent;

private javax.swing.JButton displayAllStudents;    public javax.swing.JPanel homePanel;

private javax.swing.JButton removeStudent;    private    javax.swing.JButton
updateStudentDetails;
}

```

CheckStudentPanel.java

```

package com.mycompany.studentbiodata;

import java.awt.Dimension;

import java.awt.Image;

import java.sql.*;

import java.util.logging.Level;

```

```

import java.util.logging.Logger;

import javax.swing.*;

public class CheckStudentPanel extends javax.swing.JPanel {

    public CheckStudentPanel() {

        initComponents();

    }

    private void checkBtnActionPerformed(java.awt.event.ActionEvent evt) {

        studentPhoto.setText("");

        studentPhoto.setIcon(null);

        try {

            Connection con = GetConnection.createConnection();

            String query = "SELECT * FROM studentBiodata WHERE ROLLNUMBER = ?";

            PreparedStatement statement = con.prepareStatement(query);

            statement.setString(1, rollNumberCheckField.getText());

            ResultSet rs;

            rs = statement.executeQuery();

            DefaultTableModel tModel = (DefaultTableModel) studentData.getModel();

            Object[] values = {"Name", "Roll Number", "Father Name", "Mother Name", "Date
Of Birth", "Gender", "Email", "Branch", "Attendance(%)", "CGPA", "Course Duration",
"Address"};

            for (int i = 0; i < values.length; i++) { tModel.setValueAt(values[i], i, 0);}

            for (int i = 0; i < 12; i++) {tModel.setValueAt(null, i, 1);}

            if (rs.next() != false) {

                Blob blob = rs.getBlob("photo");

                byte[] imageData = blob.getBytes(1, (int) blob.length());

                ImageIcon imageIcon = new ImageIcon(imageData);

                Image image = imageIcon.getImage();

                Image resizedImage = image.getScaledInstance(300, 300,
Image.SCALE_SMOOTH);
                studentPhoto.setIcon(new ImageIcon(resizedImage));

                studentData.setPreferredSize(new Dimension(imageIcon.getIconWidth(),
imageIcon.getIconHeight()));
            }

        } catch (SQLException ex) {
            Logger.getLogger(CheckStudentPanel.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
}

```

```

        Object[] obj = {rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4),
rs.getDate(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getDouble(9),
rs.getDouble(10), rs.getString(11), rs.getString(12)};

        for (int i = 0; i < obj.length; i++) { tModel.setValueAt(obj[i], i, 1);}
    } else {

        studentPhoto.setText("Student Not Found");

        studentPhoto.setSize(12, 12);

    }} catch (SQLException ex) {

        JOptionPane.showMessageDialog(null, ex.getMessage());

    } }

private void checkBackBtnActionPerformed(java.awt.event.ActionEvent evt) {

adminPanelRef.adminHomePanel.remove(this);adminPanelRef.adminHomePanel.add(admin
PanelRef.homePanel);

adminPanelRef.adminHomePanel.repaint();adminPanelRef.adminHomePanel.revalidate();

}

AdminPanel adminPanelRef=null;

public void AdminPanelReference(AdminPanel adminPanelRef){

    this.adminPanelRef = adminPanelRef;

}

private javax.swing.JButton checkBackBtn; private javax.swing.JButton checkBtn;

private javax.swing.JScrollPane jScrollPane1; private javax.swing.JTextField
rollNumberCheckField;

private javax.swing.JLabel rollNumberLabel; private javax.swing.JTable studentData;

private javax.swing.JLabel studentPhoto;

}

```

DisplayAllStudentPanel.java

```

package com.mycompany.studentbiodata;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

```

```

import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.table.DefaultTableModel;

public class DisplayAllStudentPanel extends javax.swing.JPanel {

    public DisplayAllStudentPanel() {
        initComponents();

        try {
            Connection con = GetConnection.createConnection();

            String query = "select * from studentBiodata";

            PreparedStatement statement = con.prepareStatement(query);

            ResultSet rs;

            rs = statement.executeQuery();

            DefaultTableModel model = (DefaultTableModel) studentDataTable.getModel();

            int i = 1;

            while (rs.next()) {

                Object[] obj = {rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4),
rs.getDate(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getDouble(9),
rs.getDouble(10), rs.getString(11), rs.getString(12)};

                model.addRow(obj);

            }

        } catch (SQLException ex) {

            Logger.getLogger(AdminPanel.class.getName()).log(Level.SEVERE, null, ex);}

        private void displayAllBackBtnActionPerformed(java.awt.event.ActionEvent evt) {

            adminPanelRef.adminHomePanel.remove(this);
            adminPanelRef.adminHomePanel.add(adminPanelRef.homePanel);

            adminPanelRef.adminHomePanel.repaint();
            adminPanelRef.adminHomePanel.revalidate();

        }

        AdminPanel adminPanelRef=null;

        public void AdminPanelReference(AdminPanel adminPanelRef){

            this.adminPanelRef = adminPanelRef;

```

```

    }

    private javax.swing.JButton displayAllBackBtn;

    private javax.swing.JScrollPane displayAllStudentScroll;

    private javax.swing.JTable studentDataTable;

}

```

RemoveStudentPanel.java

```

package com.mycompany.studentbiodata;

import java.awt.CardLayout;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class RemoveStudentPanel extends javax.swing.JPanel {

    public RemoveStudentPanel() {
        initComponents();
    }

    private void removeBtnActionPerformed(java.awt.event.ActionEvent evt) {

        String query = "delete from studentBiodata where ROLLNUMBER = ?";

        ProcessData processData = new ProcessData(query);

        boolean status = processData.removeStudent( rollNumberToRemoveField.getText());

        if (status) {
            removeStatus.setText(rollNumberToRemoveField.getText() + " removed
Successfully");
        } else {
            removeStatus.setText("No such roll number found.."); } }

    private void backBtnActionPerformed(java.awt.event.ActionEvent evt) {

        adminPanelRef.adminHomePanel.remove(this);
        adminPanelRef.adminHomePanel.add(adminPanelRef.homePanel);

        adminPanelRef.adminHomePanel.repaint();
        adminPanelRef.adminHomePanel.revalidate();

    }

    AdminPanel adminPanelRef=null;

```



```

    public void AdminPanelReference(AdminPanel adminPanelRef){
        this.adminPanelRef = adminPanelRef;
    }

    private javax.swing.JButton backBtn;
    private javax.swing.JButton removeBtn; private javax.swing.JLabel removeStatus;
    private javax.swing.JPanel removeStudentPanel; private javax.swing.JLabel rollNumber;
    private javax.swing.JTextField rollNumberToRemoveField;
}

```

UpdateStudentPanel.java

```

package com.mycompany.studentbiodata;

import java.awt.Dimension;
import java.awt.Image;
import java.io.*;
import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Enumuration;
import javax.swing.*;

public class UpdateStudentPanel extends javax.swing.JPanel {
    SetGetStudentData setGetStudentData=null;

    public UpdateStudentPanel() {
        SetGetStudentData setGetStudentData = new SetGetStudentData();
        initComponents();
    }
    boolean changePhotoStatus = false;

    private void updateBtnActionPerformed(java.awt.event.ActionEvent evt) {
        setGetStudentData.setName(studentNameField1.getText());
        setGetStudentData.setRollNumber(rollNumberField1.getText());
        setGetStudentData.setFatherName(fatherNameField1.getText());
        setGetStudentData.setMotherName(motherNameField1.getText());
        setGetStudentData.setDateOfBirth(dateOfBirthField1.getText());
    }
}

```

```

setGetStudentData.setEmail(emailIdField1.getText());
setGetStudentData.setBranch(branchField1.getText());
if(attendaceField1.getText().isEmpty()){
    setGetStudentData.setAttendancePercentage(0.0);
}else{
    setGetStudentData.setAttendancePercentage(Double.parseDouble(attendaceField1.getText()));
}

if(gpaField1.getText().isEmpty()){
    setGetStudentData.setCgpa(0.0);
}else{
    setGetStudentData.setCgpa(Double.parseDouble(gpaField1.getText()));
}

setGetStudentData.setAddress(addressField1.getText());
setGetStudentData.setCourseDuration((String) courseDurationField1.getSelectedItem());
Enumeration<AbstractButton> buttons = gender.getElements();
while (buttons.hasMoreElements()) {
    AbstractButton button = buttons.nextElement();
    if (button.isSelected()) {
        setGetStudentData.setGender(button.getText());
        break;
    }
}

ProcessData processData = new ProcessData();
if(changePhotoStatus){
    if (processData.nullCheck(setGetStudentData)) {
        String query = "UPDATE STUDENTBIODATA SET NAME = ?,
        FATHERNAME = ?, MOTHERNAME = ?, DATEOFBIRTH = ?, GENDER = ?, EMAIL
        = ?, BRANCH = ?, ATTENDANCEPERCENTAGE = ?, CGPA = ?, COURSEDURATION
        = ?, ADDRESS = ?, PHOTO = ?, WHERE ROLLNUMBER = ?";

        processData.setPreparedStatement(query);

        boolean status = processData.addStudent(setGetStudentData);

        if (status) {

```

```

        JOptionPane.showMessageDialog(null, "Data successfully saved.");
    } else {
        JOptionPane.showMessageDialog(null, "Data not saved");
    }
}
else {
    JOptionPane.showMessageDialog(null, "Enter all Fields to Save");
}
}}
else{
    if (setGetStudentData.getName() != null && setGetStudentData.getRollNumber() !=
null && setGetStudentData.getFatherName() != null &&
setGetStudentData.getMotherName() != null && setGetStudentData.getDateOfBirth() != null
&& setGetStudentData.getGender() != null && setGetStudentData.getEmail() != null &&
setGetStudentData.getBranch() != null && setGetStudentData.getAttendancePercentage() !=
0 && setGetStudentData.getCgpa() != 0 && setGetStudentData.getCourseDuration() != null
&& setGetStudentData.getAddress() != null) {
        try {
            Connection con = GetConnection.createConnection();

            String query = "UPDATE STUDENTBIODATA SET NAME = ?,
FATHERNAME = ?, MOTHERNAME = ?, DATEOFBIRTH = ?, GENDER = ?, EMAIL
= ?, BRANCH = ?, ATTENDANCEPERCENTAGE = ?, CGPA = ?, COURSEDURATION
= ?, ADDRESS = ?, WHERE ROLLNUMBER = ?";

            PreparedStatement preparedStatement = con.prepareStatement(query);

            preparedStatement.setString(1, setGetStudentData.getName());

            preparedStatement.setString(12, setGetStudentData.getRollNumber());

            preparedStatement.setString(2, setGetStudentData.getFatherName());

            preparedStatement.setString(3, setGetStudentData.getMotherName());

            preparedStatement.setString(4, setGetStudentData.getDateOfBirth());

            preparedStatement.setString(5, setGetStudentData.getGender());

            preparedStatement.setString(6, setGetStudentData.getEmail());

            preparedStatement.setString(7, setGetStudentData.getBranch());

            preparedStatement.setDouble(8, setGetStudentData.getAttendancePercentage());

            preparedStatement.setDouble(9, setGetStudentData.getCgpa());

            preparedStatement.setString(10, setGetStudentData.getCourseDuration());

            preparedStatement.setString(11, setGetStudentData.getAddress());

            int x = preparedStatement.executeUpdate();

```

```

        if (x == 1) {
            JOptionPane.showMessageDialog(null, "Data successfully saved.");
        } else {
            throw new SQLException("Error something went wrong!!!");
        } } catch (Exception s) {
            JOptionPane.showMessageDialog(null, "Unexcepected-Error");
            adminPanelRef.dispose();
            s.printStackTrace();
        } }
    else {
        JOptionPane.showMessageDialog(null, "Enter all Fields to Save");
    } } }

private void backBtnActionPerformed(java.awt.event.ActionEvent evt) {
    adminPanelRef.adminHomePanel.remove(this);
    adminPanelRef.adminHomePanel.add(adminPanelRef.homePanel);
    adminPanelRef.adminHomePanel.repaint();
    adminPanelRef.adminHomePanel.revalidate();
}

AdminPanel adminPanelRef=null;

public void AdminPanelReference(AdminPanel adminPanelRef){
    this.adminPanelRef = adminPanelRef;
}

private void checkUpdateBtnActionPerformed(java.awt.event.ActionEvent evt) {
    setGetStudentData = new SetGetStudentData();
    try {
        String rollNumber = updateRollNumberField.getText();
        Connection con = GetConnection.createConnection();
        String query = "Select * from studentbiodata where rollNumber = ?";
        PreparedStatement statement = con.prepareStatement(query);
        statement.setString(1, rollNumber);
    }
}

```

```

ResultSet rs = statement.executeQuery();

if (rs.next() != false) {

    System.out.println(rs);

    Object[] obj = {rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4),
rs.getDate(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getDouble(9),
rs.getDouble(10), rs.getString(11), rs.getString(12)};

    setGetStudentData.setName((String) obj[0]);

    setGetStudentData.setRollNumber((String) obj[1]);

    setGetStudentData.setFatherName((String) obj[2]);

    setGetStudentData.setMotherName((String) obj[3]);

    Date date = (Date) obj[4];

    String pattern = "dd-MM-yyyy";

    SimpleDateFormat sdf = new SimpleDateFormat(pattern);

    setGetStudentData.setDateOfBirth(sdf.format(date));

    setGetStudentData.setEmail((String) obj[6]);

    setGetStudentData.setBranch((String) obj[7]);

    setGetStudentData.setAttendancePercentage((double) obj[8]);

    setGetStudentData.setCgpa((double) obj[9]);

    setGetStudentData.setAddress((String) obj[11]);

    setGetStudentData.setCourseDuration((String) obj[10]);

    switch ((String)obj[5]) {

        case "Male": male1.setSelected(true); setGetStudentData.setGender("Male");

        break;

        case "Female": female1.setSelected(true);setGetStudentData.setGender("Male");

        break;

        case "others": others1.setSelected(true); setGetStudentData.setGender("Male");

        break;

    }

    Blob blob = rs.getBlob("photo");

    byte[] imageData = blob.getBytes(1, (int) blob.length());

    ImageIcon imageIcon = new ImageIcon(imageData);

```

```

        Image image = imageIcon.getImage();
        Image resizedImage = image.getScaledInstance(300, 300,
Image.SCALE_SMOOTH);
        photoStudent.setIcon(new ImageIcon(resizedImage));
        photoStudent.setPreferredSize(new Dimension(imageIcon.getIconWidth(),
imageIcon.getIconHeight()));
        studentNameField1.setText(setGetStudentData.getName());
        rollNumberField1.setText(setGetStudentData.getRollNumber());
        fatherNameField1.setText(setGetStudentData.getFatherName());
        motherNameField1.setText(setGetStudentData.getMotherName());
        dateOfBirthField1.setText(setGetStudentData.getDateOfBirth());
        emailIdField1.setText(setGetStudentData.getEmail());
        branchField1.setText(setGetStudentData.getBranch());

        attendaceField1.setText(String.valueOf(setGetStudentData.getAttendancePercentage()));
        gpaField1.setText(String.valueOf(setGetStudentData.getCgpa()));
        courseDurationField1.setSelectedItem(setGetStudentData.getCourseDuration());
        addressField1.setText(setGetStudentData.getAddress());
    }
    else{
        JOptionPane.showMessageDialog(null, "Roll Number does not exist");
        updateRollNumberField.setText(null);
    }
} catch (SQLException e) {
    e.printStackTrace();}
}

private void changePhotoBtnActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser fileChooser = new JFileChooser();
    int returnValue = fileChooser.showOpenDialog(null);
    ArrayList<String> ALLOWED_EXTENSIONS = new ArrayList<String>();
    ALLOWED_EXTENSIONS.add(".jpg"); ALLOWED_EXTENSIONS.add(".jpeg");
    ALLOWED_EXTENSIONS.add(".png"); ALLOWED_EXTENSIONS.add(".gif");
}

```

```

if (returnValue == JFileChooser.APPROVE_OPTION) {
    File selectedFile = fileChooser.getSelectedFile();
    ProcessData pd = new ProcessData();
    String extension = pd.getFileExtension(selectedFile.getAbsolutePath());
    try{
        if (!ALLOWED_EXTENSIONS.contains(extension.toLowerCase())) {
            throw new IllegalArgumentException("Photo Format Error");
        }
    }
    else{
        try {
            ImageIcon imageIcon = new ImageIcon(selectedFile.getAbsolutePath());
            Image image = imageIcon.getImage();
            Image resizedImage = image.getScaledInstance(300, 300,
Image.SCALE_SMOOTH);
            photoStudent.setIcon(new ImageIcon(resizedImage));
            changePhotoBtn.setText(selectedFile.getPath());
            uploadPhotoToDatabase(selectedFile);
            changePhotoStatus = true;
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
        }
        catch(IOException ioe){
        }
    }
    } catch(IllegalArgumentException iae){
        changePhotoBtn.setText("Change Photo");
        setGetStudentData.setPhotoPath(null);
        JOptionPane.showMessageDialog(null, "Error: " + iae.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

private void uploadPhotoToDatabase(File selectedFile) throws SQLException, IOException
{

```

```

try {
    FileInputStream studentPhotoFile = new FileInputStream(selectedFile);
    setGetStudentData.setPhotoPath(studentPhotoFile);
} catch (IOException ex) {
    ex.printStackTrace(); }
}

private javax.swing.JTextArea addressField1;      private      javax.swing.JTextField
attendanceField1;      private javax.swing.JButton backBtn1;      private
javax.swing.JTextField branchField1;      private javax.swing.JButton changePhotoBtn;
    private javax.swing.JButton checkUpdateBtn;      private
javax.swing.JComboBox<String> courseDurationField1; private      javax.swing.JTextField
dateOfBirthField1; private javax.swing.JLabel dobLabel;      private
javax.swing.JTextField emailIdField1;      private      javax.swing.JTextField
fatherNameField1; private javax.swing.JLabel fatherNameLabel;      private
javax.swing.JRadioButton female1; private javax.swing.ButtonGroup gender; private
javax.swing.JTextField gpaField1; private      javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JRadioButton male1; private      javax.swing.JTextField
motherNameField1; private javax.swing.JLabel motherNameLabel;      private
javax.swing.JRadioButton others1; private javax.swing.JLabel photoLabel;      private
javax.swing.JLabel photoStudent; private javax.swing.JLabel rollNumLabel; private
javax.swing.JTextField rollNumberField1; private javax.swing.JLabel rollNumberLabel;
    private javax.swing.JTextField studentNameField1; private      javax.swing.JLabel
studentNameLabel; private javax.swing.JButton updateBtn;      private
javax.swing.JTextField updateRollNumberField; private      javax.swing.JScrollPane
updateStudentPane; private javax.swing.JPanel updateStudentPanel1;
}

```

SetGetStudentData.java

```

package com.mycompany.studentbiodata;

import java.io.FileInputStream;

public class SetGetStudentData {

    private String name;      private String rollNumber;

    private String fatherName; private String motherName;

    private String dateOfBirth;      private String gender;

    private String email;      private String branch;

    private double attendancePercentage;      private double cgpa;

    private String courseDuration;      private String address;

    private FileInputStream photoPath;
}

```



```

public void setPhotoPath(FileInputStream photoPath){
    this.photoPath= photoPath;
}

public FileInputStream getPhotoPath(){
    return photoPath;
}

public SetGetStudentData(){
}

public SetGetStudentData(String name, String rollNumber, String fatherName, String
motherName, String dateOfBirth,
cgpa,
    String gender, String email, String branch, double attendancePercentage, double
    String courseDuration, String address) {
    this.name = name; this.rollNumber = rollNumber; this.fatherName = fatherName;
    this.motherName = motherName;      this.dateOfBirth = dateOfBirth;
    this.gender = gender;   this.email = email;
    this.branch = branch;   this.attendancePercentage = attendancePercentage;
    this.cgpa = cgpa;       this.courseDuration = courseDuration;
    this.address = address;
}

public String getName() {
    return name;}

public void setName(String name) {
    this.name = name;}

public String getRollNumber() {
    return rollNumber;}

public void setRollNumber(String rollNumber) {
    this.rollNumber = rollNumber;
}

public String getFatherName() {
    return fatherName;}

```

```

public void setFatherName(String fatherName) {
    this.fatherName = fatherName;}
public String getMotherName() {
    return motherName;}
public void setMotherName(String motherName) {
    this.motherName = motherName;
}
public String getDateOfBirth() {
    return dateOfBirth;
}
public void setDateOfBirth(String dateOfBirth) {
    this.dateOfBirth = dateOfBirth;
}
public String getGender() {
    return gender;
}
public void setGender(String gender) {
    this.gender = gender;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getBranch() {
    return branch;
}
public void setBranch(String branch) {
    this.branch = branch;
}

```

```

    }

    public double getAttendancePercentage() {
        return attendancePercentage;
    }

    public void setAttendancePercentage(double attendancePercentage) {
        this.attendancePercentage = attendancePercentage;
    }

    public double getCgpa() {
        return cgpa;
    }

    public void setCgpa(double cgpa) {
        this.cgpa = cgpa;
    }

    public String getCourseDuration() {
        return courseDuration;
    }

    public void setCourseDuration(String courseDuration) {
        this.courseDuration = courseDuration;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }
}

```

ProcessData.java

```

package com.mycompany.studentbiodata;

import java.awt.Dimension;

import java.awt.Image;

import java.sql.*;

```

```

import javax.swing.*;

public class ProcessData {
    private Connection con = null;
    private PreparedStatement preparedStatement = null;

    ProcessData(){
        con = GetConnection.createConnection();
    }

    public ProcessData(String query) {
        try {
            con = GetConnection.createConnection();
            preparedStatement = con.prepareStatement(query);
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage());} }

    public void setPreparedStatement(String query){
        try {
            preparedStatement = con.prepareStatement(query);
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage());
        } }

    public boolean removeStudent(String rollNumber) {
        try {
            preparedStatement.setString(1, rollNumber);
            int x = preparedStatement.executeUpdate();
            if (x == 0) {
                return false;
            }
        } catch (SQLException sqlException) {
            JOptionPane.showMessageDialog(null, sqlException.getMessage());
        }

        return true;
    }
}

```

```

    }

    public boolean verifyAdminDetails(String userName, String password) {
        try {
            preparedStatement.setString(1, userName);
            preparedStatement.setString(2, password);
            ResultSet s = preparedStatement.executeQuery();
            if (s.next()) {
                return true;
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
        return false;
    }

    public boolean addStudent(SetGetStudentData setGetStudentData) {
        try {
            preparedStatement.setString(1, setGetStudentData.getName());
            preparedStatement.setString(13, setGetStudentData.getRollNumber());
            preparedStatement.setString(2, setGetStudentData.getFatherName());
            preparedStatement.setString(3, setGetStudentData.getMotherName());
            preparedStatement.setString(4, setGetStudentData.getDateOfBirth());
            preparedStatement.setString(5, setGetStudentData.getGender());
            preparedStatement.setString(6, setGetStudentData.getEmail());
            preparedStatement.setString(7, setGetStudentData.getBranch());
            preparedStatement.setDouble(8, setGetStudentData.getAttendancePercentage());
            preparedStatement.setDouble(9, setGetStudentData.getCgpa());
            preparedStatement.setString(10, setGetStudentData.getCourseDuration());
            preparedStatement.setString(11, setGetStudentData.getAddress());
            preparedStatement.setBlob(12, setGetStudentData.getPhotoPath());

            System.out.println(setGetStudentData.getPhotoPath());
        }
    }

```

```

        int x = preparedStatement.executeUpdate();

        if (x == 0) {
            return false;
        }
    } catch (Exception s) {
        String errorMessage = s.getMessage();

        if (errorMessage.contains("date") || errorMessage.contains("format")) {
            JOptionPane.showMessageDialog(null, "Date formatting error occurred: " +
            errorMessage, "Error", JOptionPane.ERROR_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(null, "An error occurred: " + errorMessage,
            "Error", JOptionPane.ERROR_MESSAGE);
        }

        return false;
    }

    return true;
}

public String getFileExtension(String fileName) {
    int lastDotIndex = fileName.lastIndexOf('.');

    if (lastDotIndex == -1) {
        return "";
    }

    return fileName.substring(lastDotIndex);
}

public boolean nullCheck(SetGetStudentData setGetStudentData){
    if(setGetStudentData.getName() != null && setGetStudentData.getRollNumber() != null
    && setGetStudentData.getFatherName() != null && setGetStudentData.getDateOfBirth() != null
    && setGetStudentData.getMotherName() != null && setGetStudentData.getEmail() != null &&
    setGetStudentData.getGender() != null && setGetStudentData.getBranch() != null &&
    setGetStudentData.getAttendancePercentage() != 0 && setGetStudentData.getCgpa() != 0
    && setGetStudentData.getCourseDuration() != null && setGetStudentData.getAddress() !=
    null && setGetStudentData.getPhotoPath() != null){

        return true;
    }
}

```

```

        else

            return false;
    }

GetConnection.java

import java.sql.*;
import java.util.Properties;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

public class GetConnection {

    public static Connection createConnection() {

        Connection connection= null;

        try {

            Properties properties = new Properties();

            properties.load(new FileInputStream("d:/oracle.properties"));

            connection
            DriverManager.getConnection(properties.getProperty("url"),properties.getProperty("username"),properties.getProperty("password"));

        }

        catch(SQLException e) {

            System.out.println(e.getMessage()+" exception here");

        }

        catch(FileNotFoundException fnfe){

            System.out.print(fnfe.getMessage());

        }

        catch(IOException io){

            System.out.println(io.getMessage());

        }

        return connection;

    }
}

```

EmailVerifier.java

```
package com.mycompany.studentbiodata;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class EmailVerifier {

    private String email;

    public EmailVerifier(String email) {

        this.email = email;

    }

    public boolean verifyEmail() {

        String emailPattern = "^[a-zA-Z0-9_+&*-.]+(?:\\.[a-zA-Z0-9_+&*-.]+)*@(?:[a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,7}$";

        Pattern pattern = Pattern.compile(emailPattern);

        Matcher matcher = pattern.matcher(email);

        return matcher.matches();}}}
```

AdminData.sql

```
CREATE TABLE "SYSTEM"."ADMINDATA"

(
    "USERNAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,
    "PASSWORD" VARCHAR2(20 BYTE) NOT NULL ENABLE

);
```

StudentBioData.sql

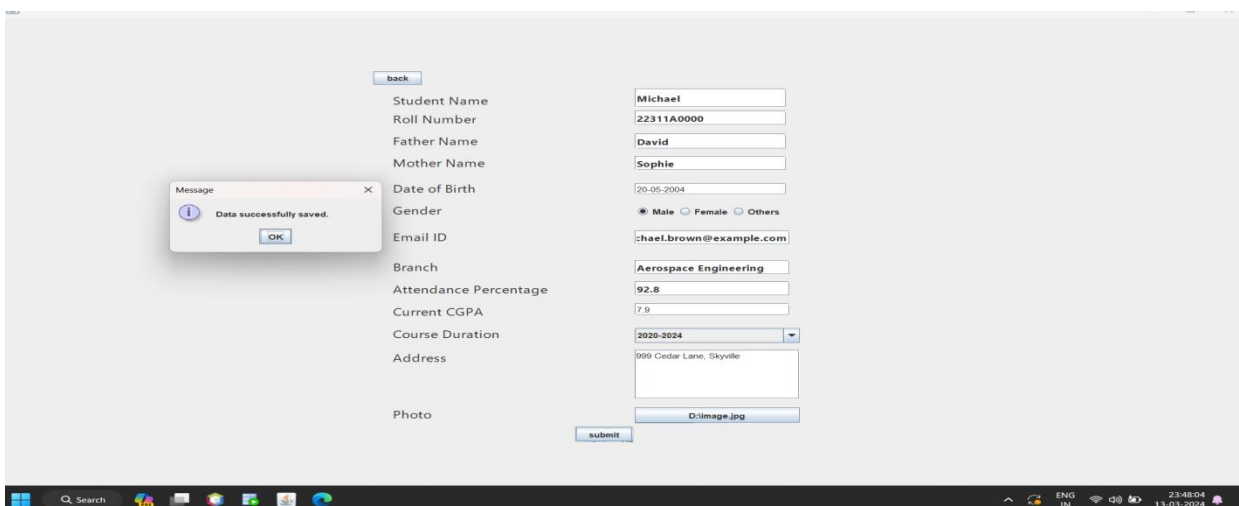
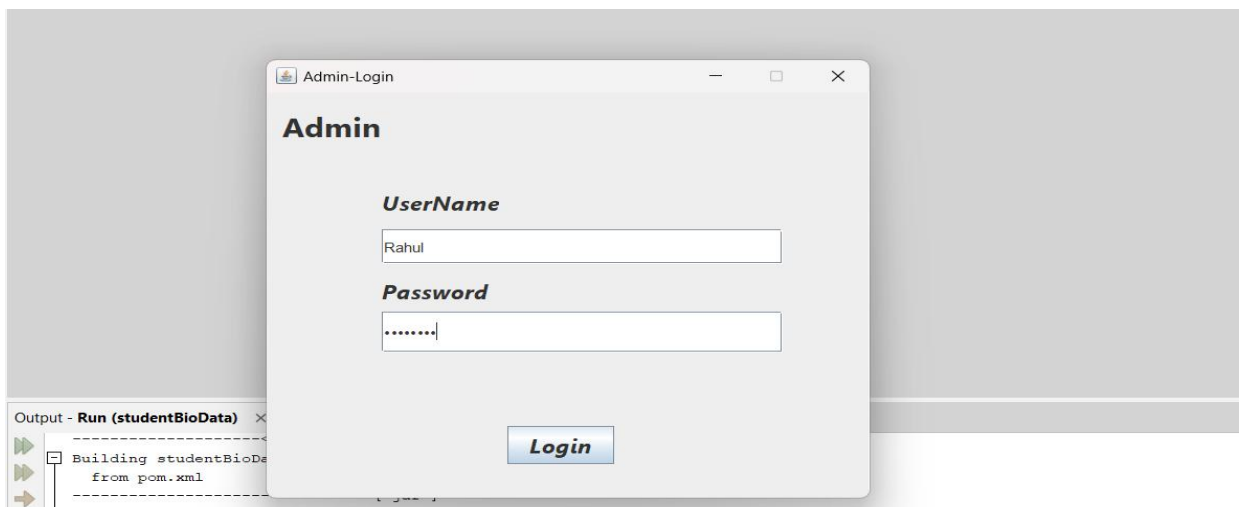
```
CREATE TABLE "SYSTEM"."STUDENTBIODATA"

(
    "NAME" VARCHAR2(255 BYTE),
    "ROLLNUMBER" VARCHAR2(20 BYTE),
    "FATHERNAME" VARCHAR2(255 BYTE),
    "MOTHERNAME" VARCHAR2(255 BYTE),
    "DATEOFBIRTH" DATE,
    "GENDER" VARCHAR2(10 BYTE),
    "EMAIL" VARCHAR2(255 BYTE),
```



```
"BRANCH" VARCHAR2(100 BYTE),  
"ATTENDANCEPERCENTAGE" NUMBER(*,0),  
"CGPA" NUMBER(*,0),  
"COURSE DURATION" VARCHAR2(30 BYTE),  
"ADDRESS" VARCHAR2(255 BYTE),  
"PHOTO" BLOB,  
PRIMARY KEY ("ROLLNUMBER")  
);
```

Output




back

Roll Number :

check

Photo




Change Photo

Student Name
Roll Number
Father Name
Mother Name
Date of Birth
Gender ☒ Male ☐ Female ☐ others
Email ID

Back

Roll Number

Check



| Details | Values |
|-----------------|----------------------------|
| Name | Michael |
| Roll Number | 22311A0000 |
| Father Name | David |
| Mother Name | Sophie |
| Date Of Birth | 2004-05-20 |
| Gender | Male |
| Email | michael.brown@example.c... |
| Branch | Aerospace Engineering |
| Attendance(%) | 93.0 |
| CGPA | 8.0 |
| Course Duration | 2020-2024 |
| Address | 999 Cedar Lane, Skyville |

Back

Roll Number

22311A0000 removed Successfully

remove

Back

Roll Number22311A0000

22311A0000 removed Successfully

remove

Back

| name | Roll Number | Father Name | Mother Name | Date Of Birth | Gender | Mail | Branch | Attendance | Cgpa | courseDuration | Address |
|----------------|-------------|---------------|----------------|---------------|--------|----------------|-----------------|------------|------|----------------|----------------|
| John Doe | 2023001001 | Michael Doe | Jennifer Doe | 2005-08-15 | Male | john.doe@ex. | Computer Sc. | 91.0 | 4.0 | 2022-2026 | 123 Main Stre. |
| Alice Johnson | 2023001003 | Bob Johnson | Karen Johnson | 2005-07-10 | Female | alice.johnson. | Mechanical E. | 92.0 | 4.0 | 2022-2026 | 789 Oak Ave. |
| Michael Wilks | 2023001004 | James Wilks | Sarah Wilks | 2005-06-25 | Male | michael.wilks. | Civil Engineer | 88.0 | 4.0 | 2022-2026 | 101 Pine Roa. |
| Emma Brown | 2023001005 | Matthew Bro. | Olivia Brown | 2005-10-30 | Female | emma.brown. | Information T. | 91.0 | 4.0 | 2022-2026 | 222 Cedar La. |
| William Jones | 2023001006 | Daniel Jones | Sophia Jones | 2005-03-15 | Male | william.jones. | Chemical En. | 90.0 | 4.0 | 2022-2026 | 333 Maple Dr. |
| Sophia Martin | 2023001007 | Alexander Ma. | Isabella Marti | 2005-01-05 | Female | sophia.martin. | Computer Sc. | 93.0 | 4.0 | 2022-2026 | 444 Oak Stre. |
| Matthew Gar... | 2023001008 | Ethan Garcia | Amelia Garcia | 2005-11-12 | Male | matthew.garc. | Electrical Eng. | 91.0 | 4.0 | 2022-2026 | 555 Birch Av. |
| Olivia Miller | 2023001009 | Noah Miller | Ava Miller | 2005-04-28 | Female | olivia.miller@ | Mechanical E. | 89.0 | 4.0 | 2022-2026 | 666 Elm Lane. |
| Ethan Taylor | 2023001010 | Liam Taylor | Chloe Taylor | 2005-02-08 | Male | ethan.taylor. | Civil Engineer | 92.0 | 4.0 | 2022-2026 | 777 Pine Lan. |

ColumnsDataModelConstraintsGrantsStatisticsTriggersFlashbackDependenciesDetailsPartitionsIndexesSQL

NAME

ROLLNUMBER

FATHERNAME

MOTHERNAME

DATEOFBIRTH

GENDER

EMAIL

BRANCH

ATTENDANCEPERCENTAGE

CGPA

COURSESDURATION

ADDRESS

PHOTO

1

John Doe

2023001001

Michael Doe

Jennifer Doe

15-08-05

Male

joh...

Compu...

91

4

2022-2026

123 M...

(BLOB)

2

Alice Johnson

2023001003

Bob Johnson

Karen Joh...

10-07-05

Female

ali...

Mecha...

92

4

2022-2026

789 O...

(BLOB)

3

Michael Williams

2023001004

James Wi...

Sarah Wil...

25-06-05

Male

mic...

Civil...

88

4

2022-2026

101 P...

(BLOB)

4

Emma Brown

2023001005

Matthew ...

Olivia Brown

30-10-05

Female

emm...

Infor...

91

4

2022-2026

222 C...

(BLOB)

5

William Jones

2023001006

Daniel J...

Sophia Jones

15-03-05

Male

wil...

Chemi...

90

4

2022-2026

333 M...

(BLOB)

6

Sophia Martinez

2023001007

Alexande...

Isabella ...

05-01-05

Female

sop...

Compu...

93

4

2022-2026

444 O...

(BLOB)

7

Matthew Garcia

2023001008

Ethan Ga...

Amelia Ga...

12-11-05

Male

mat...

Elect...

91

4

2022-2026

555 B...

(BLOB)

8

Olivia Miller

2023001009

Noah Miller

Ava Miller

28-04-05

Female

oli...

Mecha...

89

4

2022-2026

666 E...

(BLOB)

9

Ethan Taylor

2023001010

Liam Taylor

Chloe Taylor

08-02-05

Male

eth...

Civil...

92

4

2022-2026

777 P...

(BLOB)

Conclusion

In conclusion, the Student Bio Data project, developed using Java Swing API in NetBeans, presents a secure and efficient solution tailored for educational institutions. By restricting data management access to administrators, the application ensures the integrity and confidentiality of student information.

Utilizing an Oracle database for storing student data adds robustness and scalability to the system. This choice allows for seamless integration with existing databases in educational institutions, facilitating data management and retrieval processes.

The project's emphasis on security and access control guarantees that only authorized administrators can manage the student data, preventing unauthorized access and potential data breaches. This feature is essential for maintaining compliance with privacy regulations and safeguarding sensitive student information.

Furthermore, the modular design and object-oriented principles employed in the project ensure flexibility and ease of maintenance. Future enhancements, such as additional features or improvements, can be seamlessly integrated into the system without compromising its stability or performance.

