STUDENT MANAGEMENT SYSTEM

A Project Report for Industrial Internship

Submitted by

Aniket Das
Hritick Nayak
Anirban Basak
Abhijit Naru

In the partial fulfillment for the award of the degree of

B. Tech

In the

Electrical Engineering Department
at

Future Institute of Engineering and Management

at

Ardent Computech Pvt. Ltd.

ARDENT COMPUTECH PVT. LTD.
High-End Technology Training and Project

Ardent® Computech Pvt. Ltd.
Module – 132, Ground Floor, SDF Building,
Sector V, Salt Lake, Kolkata – 700091.
☎ +91 33 40073507
✉ training@ardentcollaborations.com

CERTIFICATE FROM SUPERVISOR

This is to certify that **1.ANIKET DAS 2.HRITICK NAYAK 3.ANIRBAN BASAK 4.ABHIJIT NARU** have successfully completed the project titled **STUDENT MANAGEMENT SYSTEM** under my supervision during the period from 07/06/2021 to 21/08/2021 which is in partial fulfillment of requirements for the award of the **B. Tech** degree and submitted to the Department of **ELECTRICAL ENGINEERING** of **FUTURE INSTITUTE OF ENGINEERING AND MANAGEMENT**.

_____
Signature of the Supervisor

Date:
Name of the Project Supervisor: **PALLABI SAHA**

ARDENT COMPUTECH PVT. LTD. KOLKATA

Acknowledgement

The achievement that is associated with the successful completion of any task would be incomplete without mentioning the names of those people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our deep gratitude towards our project mentor, *Ms. PALLABI SAHA* for giving such valuable suggestions, guidance and encouragement during the development of this project work.

Last but not the least we are grateful to all the faculty members of Ardent Computech Pvt. Ltd. for their support.

CONTENTS

INTRODUCTION

The project titled “Student Management System” is a Student Management software for monitoring and controlling all the details of the student. The project “Student Management System” is developed in java, which mainly focuses on basic operations in a college like adding new students, year of admission, name, gender, department, examination, etc.

“Student Management System” is a windows application written for 32-bit windows operating system, designed to keep a track of all the students. Our software is easy to use for both beginner and advanced user. It’s user interface is very user friendly.

Objective:-

Student Management System is a term for computer based system that manage the catalogue of a college or any kind of institution. The main purpose of this system is to manage student details efficiently.

Objectives of Student Management System:

- To build a system that can receive input and generate output automatically in easy way and short time.
- To build a monitoring system that is able to monitor and manage all student’s details efficiently.
- Give an opportunity to administration’s to reduce mistakes that always happen during manual method.
- To enter and preserve details of the all students details and keep them in one place. This system has some feature, they are:
    - Admission
    - Update Student
    - Update Exam
    - Display Student
    - Display Exam List

By using student management system, the operations like admission, update student, exam date and managing student details become paperless. This system provides a user-friendly data entry with easy to understand and use. It is also created to ensure that the students details are stored properly in order to maintain their security.

This system will store all the students information that consists student name, guardian name, age, gender, blood group, year of admission to the system database.

Scope:-

Due to the many problems facing with the current system by the college, we made a completely new student management system that can store all the students details in online system for managing the activities of the student details. It is totally secured system that can be accessed only by the verified user putting the user name and password.

To eliminate the issues of conventional and manual method of storing student details in a institution, the online student management system has been created. The student management system is an application based system for assisting the administration in supervising the students. The system would provide fundamental set of features for student admission, update student details, update exam details, display student details, display student's exam date, etc.

SYSTEM ANALYSIS

Identification of need:-

- A student management system is most proficient and easy to use system for managing all the processes involved in a administration in most effective ways.

- This will reduce all the manual work pressure and the whole process will be handled easily with one click.

- It will easily store the data securely so there will be no headache for any data security.

- This system can easily be handled by only one person, there will be no worry for any chances of mistake.

- User can update student's details like exam date, name, etc easily.

Feasibility Study:-

Feasibility study is the feasibility analysis or it is a measure of the software product in terms of how much beneficial product development will be for the organization in a practical point of view. Feasibility study is carried out based on many purposes to analyze whether software product will be right in terms of development, implantation, contribution of project to the organization etc.

Types of Feasibility Study :

The feasibility study mainly concentrates on bellow five mentioned areas. Among these Economic Feasibility Study is most important part of the feasibility analysis and Legal Feasibility Study is less considered feasibility analysis.

Technical Feasibility –
In Technical Feasibility current resources both hardware software along with required technology are analyzed to develop project. This technical feasibility study gives report whether there exists correct required resources and technologies which will be used for project development. Along with this,

feasibility study also analyzes technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc.

Operational Feasibility –
In Operational Feasibility degree of providing service to requirements is analyzed along with how much easy product will be to operate and maintenance after deployment. Along with this other operational scopes are determining usability of product, Determining suggested solution by software development team is acceptable or not etc.

Economic Feasibility –
In Economic Feasibility study cost and benefit of the project is analyzed. Means under this feasibility study a detail analysis is carried out what will be cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on. After that it is analyzed whether project will be beneficial in terms of finance for organization or not.

Legal Feasibility –
In Legal Feasibility study project is analyzed in legality point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. Overall it can be said that Legal Feasibility Study is study to know if proposed project confirm legal and ethical requirements.

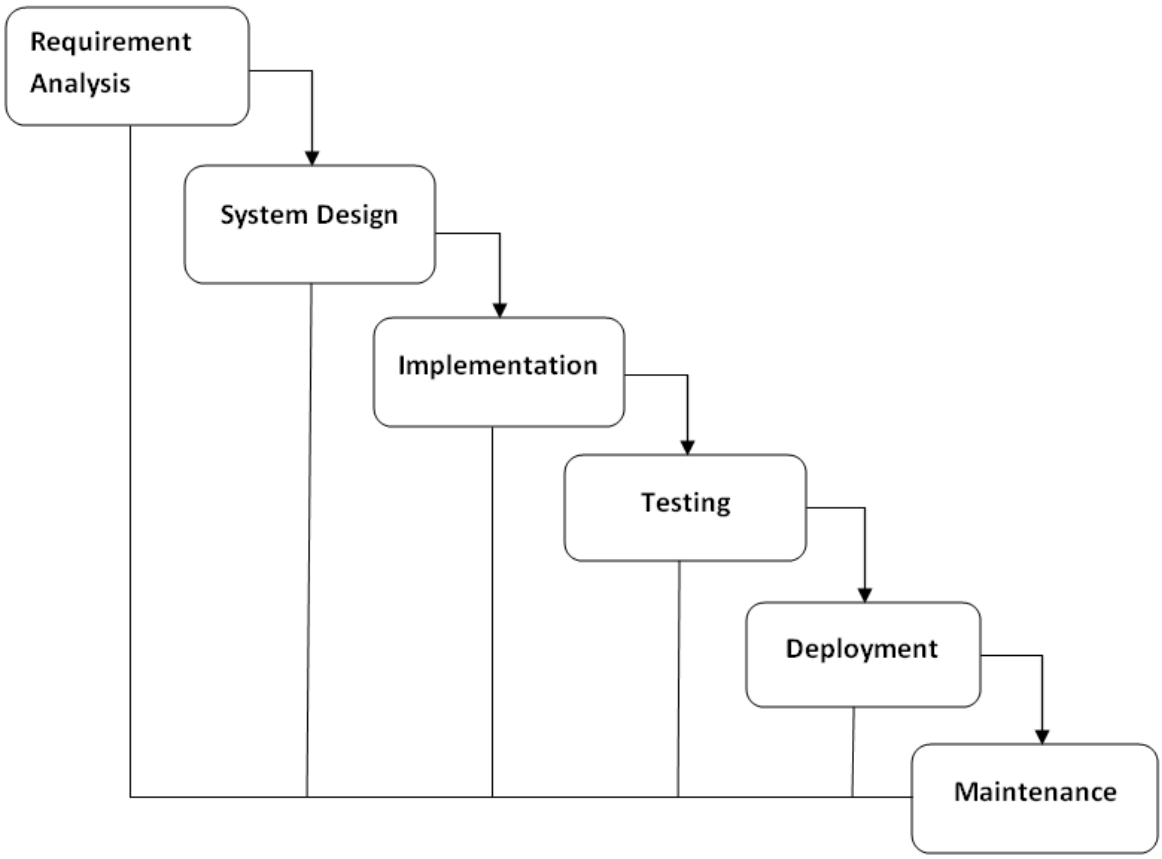
Schedule Feasibility –
In Schedule Feasibility Study mainly timelines is analyzed for proposed project which includes how much times teams will take to complete final project which has a great impact on the organization as purpose of project may fail if it can’t be completed on time.

Feasibility Study Process :
The below steps are carried out during entire feasibility analysis.

- Information assessment
- Information collection
- Report writing
- General information

Workflow Model :-

The waterfall model was the first process model to be introduced for our project .Below is the stages of the waterfall model.



It is very simple to understand and use .In a waterfall model ,each phase must be completed before the next phase can begin and there is no overlapping in the phases .Waterfall model is the earliest SDLC approach that was used for development.

Functional & Non- functional requirement:-

Functional Requirement	Non-Functional Requirement
• User can login into their account with unique user id and password. • Admin should be able to register a new student. • Admin should be able to update information about student. • Admin should be able to update exam related information.	• The system shall allow the user to access the system from any browser, no special training is required. The system should be user friendly and is written in simple English. • The system is available 100% for the user and is used by all the times. • The system should accurately provide real time information taking into consideration various issues. The system shall provide 100% reliability. • The information is refreshed at regular intervals. The system shall respond the member as soon as possible. • System will use a secured database. • For safety issues it must have two servers, one main server and one backup server. • System shall handle expected and unexpected error and should be able to handle large amount of data.

HARDWARE AND SOFTWARE REQUIREMENTS:-

HARDWARE REQUIREMENTS	SOFTWARE REQUIREMENTS
• Computer has 1Ghz or faster processor • Minimum 2GB of RAM for work efficiency • HDD 10GB Hard disk space or above	• Windows 7 or above • Notepad++ • XAMPP • Intellij IDEA

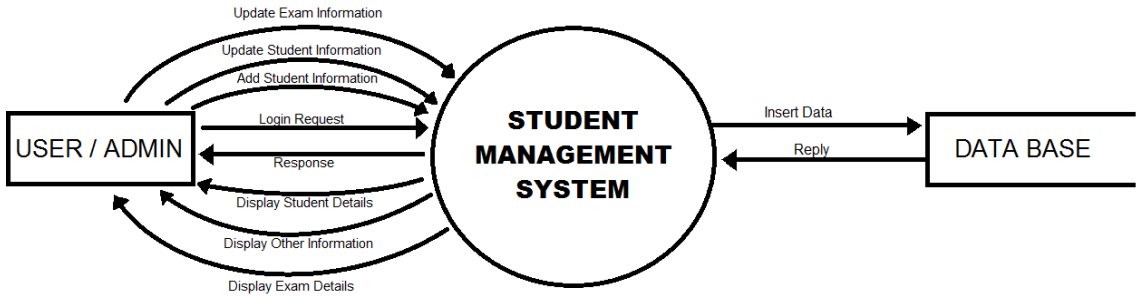
SYSTEM DESIGN

DFD:-

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes**.**

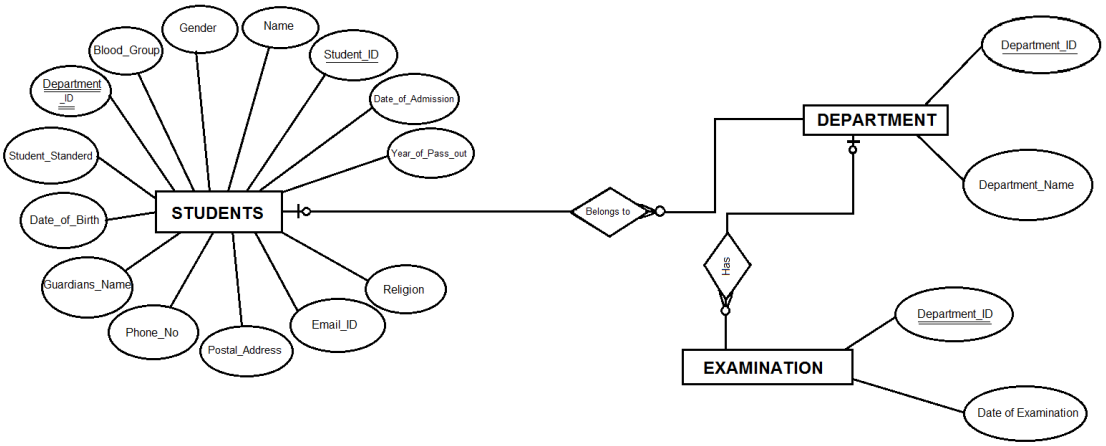
0-level DFD:

It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.



ERD:-

ERD stands for Entity Relationship Diagram. An entity relationship diagram, also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology system.



DATABASE DESIGN:-

Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept of designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

SQL Commands

Admin Log Table:

```sql
CREATE TABLE adminlog(
    adminName VARCHAR(20) PRIMARY KEY NOT NULL,
    adminPassword VARCHAR(20) NOT NULL
);
```

Department Table:

```sql
CREATE TABLE department(
    DepartmentID VARCHAR(4) PRIMARY KEY,
    DepartmentName VARCHAR(30) NOT NULL UNIQUE
);
```

Examination Table:

```sql
CREATE TABLE examination(
    DepartmentID VARCHAR(4) UNIQUE,
    FOREIGN KEY(DepartmentID) REFERENCES department(DepartmentID),
    DateOfExamination VARCHAR(11) NOT NULL
);
```

Student Table:

```sql
CREATE TABLE student(
    StudentID INT PRIMARY KEY,
    StudentName VARCHAR(50) NOT NULL,
    StudentGender VARCHAR(7) NOT NULL,
    BloodGrp VARCHAR(3) NOT NULL,
    DepartmentID VARCHAR(30) NOT NULL,
    FOREIGN KEY(DepartmentID) REFERENCES department(DepartmentID),
    StudyStandard VARCHAR(5) NOT NULL,
    DOB VARCHAR(11) NOT NULL,
    GuardianName VARCHAR(50) NOT NULL,
    PhoneNumber VARCHAR(12) NOT NULL UNIQUE,
    PostalAddress VARCHAR(100) NOT NULL,
    EmailAddress VARCHAR(50) NOT NULL UNIQUE,
    Religion VARCHAR(12) NOT NULL,
    DateOfAdmission VARCHAR(11) NOT NULL,
    YearOfPassOut INT NOT NULL
);
```

Admin Log Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	adminName	varchar(20)	utf8mb4_0900_ai_ci		No	None			Change Drop More
2	adminPassword	varchar(20)	utf8mb4_0900_ai_ci		No	None			Change Drop More

Department Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	DepartmentID	varchar(4)	utf8mb4_0900_ai_ci		No	None			Change Drop More
2	DepartmentName	varchar(50)	utf8mb4_0900_ai_ci		No	None			Change Drop More

Examination Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	DepartmentID	varchar(4)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
2	DateOfExamination	varchar(11)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More

Student Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	StudentID	int			No	None			Change Drop More
2	StudentName	varchar(50)	utf8mb4_0900_ai_ci		No	None			Change Drop More
3	StudentGender	varchar(7)	utf8mb4_0900_ai_ci		No	None			Change Drop More
4	BloodGrp	varchar(3)	utf8mb4_0900_ai_ci		No	None			Change Drop More
5	DepartmentID	varchar(30)	utf8mb4_0900_ai_ci		No	None			Change Drop More
6	StudyStandard	varchar(4)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
7	DOB	varchar(11)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
8	GuardianName	varchar(50)	utf8mb4_0900_ai_ci		No	None			Change Drop More
9	PhoneNumber	varchar(10)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
10	PostalAddress	varchar(100)	utf8mb4_0900_ai_ci		No	None			Change Drop More
11	EmailAddress	varchar(50)	utf8mb4_0900_ai_ci		No	None			Change Drop More
12	Religion	varchar(12)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
13	DateOfAdmission	varchar(11)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
14	YearOfPassOut	int			No	None			Change Drop More

CODING AND IMPLEMENTATION

CODES:-

Log in screen:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class LogInScreen {

    Font font1 = new Font(Font.MONOSPACED,Font.BOLD, size: 30);
    Font font2 = new Font(Font.DIALOG, Font.PLAIN, size: 15);
    Font font3 = new Font(Font.DIALOG_INPUT,Font.BOLD, size: 13);

    JFrame frame1 = new JFrame();
    private JLabel titleLabel, userNameLabel, passwordLabel;
    private JTextField userNameTF;
    private JPasswordField passwordTF;
    private JButton registerButton, logInButton;

    LogInScreen(){

        frame1.setLayout(null);


        titleLabel = new JLabel( text: "Log In");
        userNameLabel = new JLabel( text: "User Name : ");
        passwordLabel = new JLabel( text: "Password : ");
        userNameTF = new JTextField();
        passwordTF = new JPasswordField();
        registerButton = new JButton( text: "Register Now");
        logInButton = new JButton( text: "Log In");
```

```
        titleLabel.setFont(font1);
        userNameLabel.setFont(font2);
        userNameTF.setFont(font2);
        passwordLabel.setFont(font2);
        passwordTF.setFont(font2);
        logInButton.setFont(font3);
        registerButton.setFont(font3);

        titleLabel.setForeground(new Color( r: 250, g: 250, b: 250));
        userNameLabel.setForeground(new Color( r: 250, g: 250, b: 250));
        passwordLabel.setForeground(new Color( r: 250, g: 250, b: 250));


        titleLabel.setBounds( x: 60, y: 50, width: 200, height: 100);
        userNameLabel.setBounds( x: 60, y: 130, width: 150, height: 30);
        userNameTF.setBounds( x: 160, y: 130, width: 250, height: 30);
        passwordLabel.setBounds( x: 60, y: 170, width: 150, height: 30);
        passwordTF.setBounds( x: 160, y: 170, width: 250, height: 30);
        registerButton.setBounds( x: 90, y: 220, width: 130, height: 30);
        logInButton.setBounds( x: 180,  y: 220,  width: 130, height: 30);

        frame1.add(titleLabel);
        frame1.add(userNameLabel);
        frame1.add(passwordLabel);
        frame1.add(userNameTF);
        frame1.add(passwordTF);
        frame1.add(logInButton);
        frame1.getContentPane().setBackground(new Color( r: 100, g: 00, b: 250));
        frame1.setSize( width: 500, height: 400);//400 width and 500 height
        frame1.setLayout(null);//using no layout managers
        frame1.setVisible(true);//making the frame visible
        frame1.setResizable(false);
        frame1.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
```

```java
        logInButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                verification(userNameTF.getText(), String.valueOf(passwordTF.getPassword()));
            }
        });

    }

    public static void main(String[] args) { new LogInScreen(); }

    protected void verification(String name, String password){

        DBConnector db = new DBConnector();
        Connection con = db.connect();

        String sql = "SELECT * FROM adminlog WHERE adminPassword = '"+ password +"' AND adminName = '"+name + "'";

        try{
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery(sql);
            if (rs.next()){
                frame1.dispose();
                new MenuFrame(String.valueOf(userNameTF.getText()));

            }else {
                JOptionPane.showMessageDialog( parentComponent: null, message: "not verified");
            }
        }catch (SQLException e){
            e.printStackTrace();
        }

    }
}
```

Menu Frame:

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MenuFrame implements ActionListener {

    Font font1 = new Font(Font.MONOSPACED,Font.BOLD, size: 25);
    Font font2 = new Font(Font.SANS_SERIF,Font.BOLD, size: 17);
    Font font3 = new Font(Font.DIALOG_INPUT,Font.BOLD, size: 13);

    JFrame frame2 = new JFrame();

    private JLabel showAdmin;
    private JButton admissionBtn,  updateStudentBtn, examDateUpdateBtn, showStudentBtn, showExamDateBtn,logOutBtn;
    private  String name;

    MenuFrame(String adminName){

        showAdmin = new JLabel( text: "YOU LOGGED IN AS : " + adminName);
        name = adminName;

        admissionBtn = new JButton( text: "Admission");
        updateStudentBtn = new JButton( text: "Update Student");
        examDateUpdateBtn = new JButton( text: "Update Exam ");
        showStudentBtn = new JButton( text: "Display Student");
        showExamDateBtn = new JButton( text: "Display Exam List");
        logOutBtn = new JButton( text: "Log Out");

        showAdmin.setFont(font1);
        admissionBtn.setFont(font2);
        updateStudentBtn.setFont(font2);
        examDateUpdateBtn.setFont(font2);
        showStudentBtn.setFont(font2);
```

```java
        showExamDateBtn.setFont(font2);
        logOutBtn.setFont(font2);

        admissionBtn.setBackground(Color.yellow);
        updateStudentBtn.setBackground(Color.yellow);
        examDateUpdateBtn.setBackground(Color.yellow);
        showStudentBtn.setBackground(Color.yellow);
        showExamDateBtn.setBackground(Color.yellow);
        logOutBtn.setBackground(Color.cyan);

        showAdmin.setForeground(Color.yellow);
        admissionBtn.setForeground(Color.RED);
        updateStudentBtn.setForeground(Color.RED);
        examDateUpdateBtn.setForeground(Color.RED);
        showStudentBtn.setForeground(Color.RED);
        showExamDateBtn.setForeground(Color.RED);
        logOutBtn.setForeground(Color.RED);

        showAdmin.setBounds( x: 10, y: 10, width: 700, height: 40);
        admissionBtn.setBounds( x: 15, y: 100, width: 180, height: 60);
        updateStudentBtn.setBounds( x: 215, y: 100, width: 180, height: 60);
        examDateUpdateBtn.setBounds( x: 415, y: 100, width: 180, height: 60);
        showStudentBtn.setBounds( x: 80, y: 210, width: 180, height: 60);
        showExamDateBtn.setBounds( x: 290, y: 210, width: 180, height: 60);
        logOutBtn.setBounds( x: 450, y: 300, width: 100, height: 50);

        admissionBtn.addActionListener( l: this);
        updateStudentBtn.addActionListener( l: this);
        examDateUpdateBtn.addActionListener( l: this);
        showStudentBtn.addActionListener( l: this);
        showExamDateBtn.addActionListener( l: this);
        logOutBtn.addActionListener( l: this);

        frame2.add(showAdmin);
```

```
        frame2.add(admissionBtn);
        frame2.add(showStudentBtn);
        frame2.add(updateStudentBtn);
        frame2.add(examDateUpdateBtn);
        frame2.add(showExamDateBtn);
        frame2.add(logOutBtn);

        frame2.setSize( width: 625, height: 400); //400 width and 500 height
        frame2.getContentPane().setBackground(new Color( r: 100, g: 00, b: 250));
        frame2.setLayout(null);//using no layout managers
        frame2.setVisible(true);//making the frame visible
        frame2.setResizable(false);
        frame2.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

    }

    @Override
    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == admissionBtn){
            new AdmissionFrame(name);
            frame2.dispose();
        } else if ( e.getSource() == showStudentBtn ){
            new DisplayStudentFrame(name);
            frame2.dispose();
        } else if ( e.getSource() == updateStudentBtn ){
            new UpdateStudentFrame(name);
            frame2.dispose();
        } else if ( e.getSource() == examDateUpdateBtn ){
            new UpdateExamFrame(name);
            frame2.dispose();
        } else if ( e.getSource() == showExamDateBtn ) {
            new DisplayExamFrame(name);
            frame2.dispose();
```

```
    }

    @Override
    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == admissionBtn){
            new AdmissionFrame(name);
            frame2.dispose();
        } else if ( e.getSource() == showStudentBtn ){
            new DisplayStudentFrame(name);
            frame2.dispose();
        } else if ( e.getSource() == updateStudentBtn ){
            new UpdateStudentFrame(name);
            frame2.dispose();
        } else if ( e.getSource() == examDateUpdateBtn ){
            new UpdateExamFrame(name);
            frame2.dispose();
        } else if ( e.getSource() == showExamDateBtn ) {
            new DisplayExamFrame(name);
            frame2.dispose();
        } else if (e.getSource() == logOutBtn){
            frame2.dispose();
            JOptionPane.showMessageDialog( parentComponent: null, message: "Logged Out");
            new LogInScreen();
        }

    }
}
```

Admission Frame:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class AdmissionFrame {


    JFrame f1 = new JFrame();
    private JLabel l, l0, l00, l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11, l12, l13, l14;
    private JTextField tf1, tf2, tf3, tf8, tf9;
    private JComboBox c1, c2, c3, cd4, cm4, cy4, c5, c6, cd7, cm7, cy7, c8;
    private JButton b1;
    private TextArea ta1;
    int no2;



    AdmissionFrame(String adminName) {

        l = new JLabel( text: "You are Logged in as : " + adminName);
        Font h = new Font(Font.DIALOG_INPUT, Font.BOLD,  size: 30);
        l.setBounds( x: 30,  y: -225,  width: 600,  height: 500);
        l.setFont(h);
        l.setForeground(Color.yellow);
```

```
        l0 = new JLabel( text: "ENTER STUDENT DETAILS");
        l0.setForeground(new Color( r: 0,  g: 191,  b: 255));
        Font h0 = new Font(Font.SANS_SERIF, Font.BOLD,  size: 40);
        l0.setFont(h0);
        l0.setBounds( x: 420,  y: -180,  width: 600,  height: 500);

        l00 = new JLabel( text: "ENTER STUDENT DETAILS");
        l00.setForeground(new Color( r: 0,  g: 255,  b: 255));
        l00.setFont(h0);
        l00.setBounds( x: 423,  y: -178,  width: 600,  height: 500);

        l1 = new JLabel( text: "Student ID : ");
        l1.setForeground(new Color( r: 153,  g: 255,  b: 255));
        Font h1 = new Font(Font.SANS_SERIF, Font.BOLD,  size: 25);
        l1.setFont(h1);
        l1.setBounds( x: 125,  y: 129,  width: 350,  height: 35);
        tf1 = new JTextField();
        Font h2 = new Font(Font.SERIF, Font.HANGING_BASELINE,  size: 25);
        tf1.setFont(h2);
        tf1.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf1.setBounds( x: 290,  y: 130,  width: 310,  height: 35);

        tf1.setText(String.valueOf(new LastEnteredStudentID().verification() + 1));
        tf1.setEditable(false);

        l2 = new JLabel( text: "Student Name : ");
        l2.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l2.setFont(h1);
        l2.setBounds( x: 700,  y: 126,  width: 350,  height: 35);
        tf2 = new JTextField();
        tf2.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf2.setFont(h2);
        tf2.setBounds( x: 900,  y: 128,  width: 310,  height: 35);

        l3 = new JLabel( text: "Guardians Name : ");
        l3.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l3.setFont(h1);
        l3.setBounds( x: 50,  y: 200,  width: 350,  height: 35);
        tf3 = new JTextField();
        tf3.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf3.setFont(h2);
        tf3.setBounds( x: 290,  y: 200,  width: 310,  height: 35);

        l4 = new JLabel( text: "Blood Group : ");
        l4.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l4.setFont(h1);
        l4.setBounds( x: 715,  y: 195,  width: 350,  height: 35);
        String country6[] = { "Choose your Blood Group", "A+", "O+", "B+", "AB+", "A-", "O-", "B-", "AB-" };
        c5 = new JComboBox(country6);
        c5.setFont(h2);
        c5.setBounds( x: 900,  y: 195,  width: 310,  height: 35);

        l5 = new JLabel( text: "Department ID : ");
        l5.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l5.setFont(h1);
        l5.setBounds( x: 75,  y: 275,  width: 350,  height: 35);
        String country1[] = { "Choose your Department", "CE", "CSE", "EE", "ECE", "IT", "ME" };
        c2 = new JComboBox(country1);
        c2.setFont(h2);
        c2.setBounds( x: 290,  y: 275,  width: 310,  height: 35);

        l6 = new JLabel( text: "Gender : ");
        l6.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l6.setFont(h1);
        l6.setBounds( x: 160,  y: 345,  width: 350,  height: 35);
        String country[] = { "Choose Gender", "Male", "Female", "Others" };
        c1 = new JComboBox(country);
        c1.setFont(h2);
        c1.setBounds( x: 290,  y: 345,  width: 310,  height: 35);

        l7 = new JLabel( text: "Studing Standard : ");
        l7.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l7.setFont(h1);
        l7.setBounds( x: 660,  y: 268,  width: 350,  height: 35);
        String country2[] = { "Choose your Year", "1 st", "2nd", "3rd", "4th" };
        c3 = new JComboBox(country2);
        c3.setFont(h2);
        c3.setBounds( x: 900,  y: 268,  width: 310,  height: 35);

        l8 = new JLabel( text: "Date Of Birth : ");
        l8.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l8.setFont(h1);
        l8.setBounds( x: 710,  y: 340,  width: 350,  height: 35);
        String country3[] = { "DD", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14",
                "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" };
        cd4 = new JComboBox(country3);
        cd4.setFont(h2);
```

```
        cd4.setBounds( x: 900,  y: 340,  width: 90,  height: 35);
        String country4[] = { "MM", "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov",
                "Dec" };
        cm4 = new JComboBox(country4);
        cm4.setFont(h2);
        cm4.setBounds( x: 990,  y: 340,  width: 100,  height: 35);
        String country5[] = { "YYYY", "1995", "1996", "1997", "1998", "1999", "2000", "2001", "2002", "2003", "2004",
                "2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017",
                "2018", "2019", "2020", "2021", "2022", "2023", "2024", "2025", "2026", "2027", "2028", "2029",
                "2030" };
        cy4 = new JComboBox(country5);
        cy4.setFont(h2);
        cy4.setBounds( x: 1090,  y: 340,  width: 120,  height: 35);

        l9 = new JLabel( text: "Email ID : ");
        l9.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l9.setFont(h1);
        l9.setBounds( x: 765,  y: 410,  width: 350,  height: 35);
        tf8 = new JTextField();
        tf8.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf8.setFont(h2);
        tf8.setBounds( x: 900,  y: 410,  width: 310,  height: 35);

        l10 = new JLabel( text: "Phone Number : ");
        l10.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l10.setFont(h1);
        l10.setBounds( x: 70,  y: 410,  width: 350,  height: 35);
        tf9 = new JTextField();
        tf9.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf9.setFont(h2);
        tf9.setBounds( x: 290,  y: 410,  width: 310,  height: 35);

        l11 = new JLabel( text: "Postal Address : ");
        l11.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l11.setFont(h1);
        l11.setBounds( x: 70,  y: 505,  width: 210,  height: 35);
        ta1 = new TextArea();
        ta1.setBackground(new Color( r: 220,  g: 220,  b: 250));
        ta1.setFont(h2);
        ta1.setBounds( x: 290,  y: 480,  width: 310,  height: 100);

        l12 = new JLabel( text: "Date of Addmission : ");
        l12.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l12.setFont(h1);
        l12.setBounds( x: 25,  y: 615,  width: 350,  height: 35);
        String country7[] = { "DD", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14",
                "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" };
        cd7 = new JComboBox(country7);
        cd7.setFont(h2);
        cd7.setBounds( x: 290,  y: 615,  width: 90,  height: 35);
        String country8[] = { "MM", "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov",
                "Dec" };
        cm7 = new JComboBox(country8);
        cm7.setFont(h2);
        cm7.setBounds( x: 380,  y: 615,  width: 100,  height: 35);
        String country9[] = { "YYYY", "2013", "2014", "2015", "2016", "2017", "2018", "2019", "2020", "2021", "2022",
                "2023", "2024", "2025", "2026", "2027", "2028", "2029", "2030" };
        cy7 = new JComboBox(country9);
        cy7.setFont(h2);
        cy7.setBounds( x: 480,  y: 615,  width: 120,  height: 35);

        l13 = new JLabel( text: "Year of Passout : ");
        l13.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l13.setFont(h1);
        l13.setBounds( x: 675,  y: 485,  width: 350,  height: 35);
        String country10[] = { "YYYY", "2016", "2017", "2018", "2019", "2020", "2021", "2022", "2023", "2024", "2025",
                "2026", "2027", "2028", "2029", "2030", "2031", "2032", "2033", "2034" };
        c6 = new JComboBox(country10);
        c6.setFont(h2);
        c6.setBounds( x: 900,  y: 485,  width: 310,  height: 35);

        l14 = new JLabel( text: "Religion : ");
        l14.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l14.setFont(h1);
        l14.setBounds( x: 770,  y: 560,  width: 350,  height: 35);
        // tf13 = new JTextField();
        // tf13.setBackground(new Color(220, 220, 250));
        // tf13.setFont(h2);
        // tf13.setBounds(900, 560, 310, 35);
        String country11[] = { "Choose Your Religion", "Hinduism", "Islam", "Cristianity", "Buddhism", "Others" };
        c8 = new JComboBox(country11);
        c8.setFont(h2);
        c8.setBounds( x: 900,  y: 560,  width: 310,  height: 35);

        b1 = new JButton( text: "SUBMIT");
        b1.setBounds( x: 1000,  y: 640,  width: 100,  height: 35);
```

```
        f1.setLayout(null);

        f1.add(l);
        f1.add(l0);
        f1.add(l00);
        f1.add(l1);
        f1.add(tf1);
        f1.add(l2);
        f1.add(tf2);
        f1.add(l3);
        f1.add(tf3);
        f1.add(l4);
        f1.add(c5);
        f1.add(l5);
        f1.add(c2);
        f1.add(c1);
        f1.add(l6);
        f1.add(c1);
        f1.add(l7);
        f1.add(c3);
        f1.add(l8);
        f1.add(cd4);
        f1.add(cm4);
        f1.add(cy4);
        f1.add(l9);
        f1.add(tf8);
        f1.add(l10);
        f1.add(tf9);
        f1.add(l11);
        f1.add(ta1);
        f1.add(l12);
        f1.add(cd7);
        f1.add(cm7);
        f1.add(cy7);
        f1.add(l13);
        f1.add(có);
        f1.add(l14);
        // f1.add(tf13);
        f1.add(c8);
        f1.add(b1);

        f1.getContentPane().setBackground(new Color( r: 100,  g: 00,  b: 250));
        f1.setSize( width: 1365,  height: 770);
        f1.setLayout(null);
        f1.setVisible(true);
        f1.setResizable(false);
        f1.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

        b1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (e.getSource() == b1) {
                    new MenuFrame(adminName);
                    f1.dispose();
                    insertion();
                }
            }
        });
    }

    protected void insertion() {

        DBConnector con = new DBConnector();
        Connection cc = con.connect();

        no2 = Integer.parseInt(String.valueOf(có.getItemAt(có.getSelectedIndex())));


        String sql = " INSERT INTO `student`(`StudentID`, `StudentName`, `StudentGender`, `BloodGrp`, `DepartmentID`, `StudyStandard`, `DOB`," +
                " `GuardianName`, `PhoneNumber`, `PostalAddress`, `EmailAddress`, `Religion`, `DateOfAdmission`, `YearOfPassOut`) VALUES ("
                + Integer.parseInt(tf1.getText()) + ", '" + String.valueOf(tf2.getText()) + "', '"
                + String.valueOf(c1.getItemAt(c1.getSelectedIndex())) + "', '"
                + String.valueOf(c5.getItemAt(c5.getSelectedIndex())) + "', '"
                + String.valueOf(c2.getItemAt(c2.getSelectedIndex())) + "', '"
                + String.valueOf(c3.getItemAt(c3.getSelectedIndex())) + "', '"
                + String.valueOf(cd4.getSelectedItem() + "/" + cm4.getSelectedItem() + "/" + cy4.getSelectedItem())
                + "', '" + String.valueOf(tf3.getText()) + "', " + Long.parseLong(tf9.getText()) + ", '"
                + String.valueOf(ta1.getText()) + "', '" + String.valueOf(tf8.getText()) + "', '"
                + String.valueOf(c8.getItemAt(c8.getSelectedIndex())) + "', '"
                + String.valueOf(cd7.getSelectedItem() + "/" + cm7.getSelectedItem() + "/" + cy7.getSelectedItem())
                + "', " + no2 + ")";
        System.out.println(sql);

        PreparedStatement ps;
        try {

            ps = cc.prepareStatement(sql);
```

```
            int rs = ps.executeUpdate();


            if (rs > 0) {
                JOptionPane.showMessageDialog( parentComponent: null,  message: "Student has been registered");

            } else {
                JOptionPane.showMessageDialog( parentComponent: null,  message: "Something went wrong, try again");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }


}
```

Last Entered Student ID:

```
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

public class LastEnteredStudentID {
    ArrayList<Integer> arrl = new ArrayList<>();
    protected int verification() {
        int l = 0;
        if(arrl.size()>1){
            arrl.clear();
        }
        DBConnector db = new DBConnector();
        Connection con = db.connect();
        String sql = "SELECT MAX(StudentID) FROM `student`";
        try {
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery(sql);
            if (rs.next()) {
                System.out.println("Maximum Student's ID : " + rs.getInt( columnLabel: "MAX(StudentID)"));
                l = rs.getInt( columnLabel: "MAX(StudentID)");
                for(int i = 1; i <= rs.getInt( columnLabel: "MAX(StudentID)"); i++){
                    arrl.add(i);
                }
                System.out.println(arrl);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return l;
    }
}
```

Update Student:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.PreparedStatement;

public class UpdateStudentFrame {
    JFrame f0, f1;
    private JLabel l10, l13, l12, l11, l, l0, l00, l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11, l12, l13, l14;
    private JTextField tf1, tf2, tf3, tf4, tf7, tf8, tf9, tf10, tf11, tf12, tf13, tf14;
    private JComboBox c0, c1, c2, c3, cm4, cm4, cy4, c5, c6, cd7, cm7, cy7, c8;
    private JButton b0, b1, b2;
    private TextArea ta1;
    int n;
    UpdateStudentFrame(String adminName) {
        f0 = new JFrame();
        l10 = new JLabel( text: "You are Logged in as : " + adminName);
        Font h = new Font(Font.DIALOG_INPUT, Font.BOLD,  size: 30);
        l10.setForeground(new Color( r: 255,  g: 215,  b: 0));
        l10.setBounds( x: 30,  y: -225,  width: 600,  height: 500);
        l10.setFont(h);
        l11 = new JLabel( text: "UPDATE STUDENT DETAILS");
        l13.setForeground(new Color( r: 0,  g: 191,  b: 255));
        Font h0 = new Font(Font.SANS_SERIF, Font.BOLD,  size: 40);
        l11.setFont(h0);
        l11.setBounds( x: 420,  y: -165,  width: 600,  height: 500);
        l12 = new JLabel( text: "UPDATE STUDENT DETAILS");
        l12.setForeground(new Color( r: 0,  g: 255,  b: 255));
        l12.setFont(h0);
        l12.setBounds( x: 423,  y: -163,  width: 600,  height: 500);
        l13 = new JLabel( text: "Enter Student's ID : ");
        l13.setForeground(new Color( r: 153,  g: 255,  b: 255));
        Font h1 = new Font(Font.SANS_SERIF, Font.BOLD,  size: 25);
        l13.setFont(h1);
        l13.setBounds( x: 350,  y: 180,  width: 350,  height: 35);
        String country0[] = new String[new LastEnteredStudentID().verification() + 1];
        country0[0] = "Choose Student's ID";
        for (int i = 1; i <= new LastEnteredStudentID().verification(); i++) {
            country0[i] = String.valueOf(i);
        }
        c0 = new JComboBox(country0);
        c0.setFont(h1);
        c0.setBounds( x: 605,  y: 180,  width: 340,  height: 35);
        b0 = new JButton( text: "UPDATE STUDENT");
        b0.setBounds( x: 570,  y: 500,  width: 200,  height: 40);
        f0.add(l10);
        f0.add(l11);
        f0.add(l12);
        f0.add(l13);
        f0.add(c0);
        f0.add(b0);
        f0.getContentPane().setBackground(new Color( r: 100,  g: 00,  b: 250));
        f0.setSize( width: 1365,  height: 770);
        f0.setLayout(null);
        f0.setVisible(true);
        f0.setResizable(false);
        f0.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        f1 = new JFrame();
        l = new JLabel( text: "You are Logged in as : " + adminName);
        l.setForeground(new Color( r: 255,  g: 215,  b: 0));
        l.setBounds( x: 30,  y: -225,  width: 600,  height: 500);
        l.setFont(h);
        l0 = new JLabel( text: "ENTER STUDENT DETAILS");
        l0.setForeground(new Color( r: 0,  g: 191,  b: 255));
        l0.setFont(h0);
        l0.setBounds( x: 420,  y: -170,  width: 600,  height: 500);
        l00 = new JLabel( text: "ENTER STUDENT DETAILS");
        l00.setForeground(new Color( r: 0,  g: 255,  b: 255));
        l00.setFont(h0);
        l00.setBounds( x: 423,  y: -168,  width: 600,  height: 500);
        l1 = new JLabel( text: "Student ID : ");
        l1.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l1.setFont(h1);
        l1.setBounds( x: 125,  y: 129,  width: 350,  height: 35);
        tf1 = new JTextField();
        Font h2 = new Font(Font.SERIF, Font.HANGING_BASELINE,  size: 25);
        tf1.setFont(h2);
        tf1.setBackground(new Color( r: 163,  g: 160,  b: 160,  a: 255));
        tf1.setBounds( x: 290,  y: 130,  width: 310,  height: 35);
        tf1.setEditable(false);
        l2 = new JLabel( text: "Student Name : ");
        l2.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l2.setFont(h1);
        l2.setBounds( x: 700,  y: 126,  width: 350,  height: 35);
        tf2 = new JTextField();
        tf2.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf2.setFont(h2);
        tf2.setBounds( x: 900,  y: 128,  width: 310,  height: 35);
        l3 = new JLabel( text: "Guardians Name : ");
        l3.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l3.setFont(h1);
        l3.setBounds( x: 50,  y: 200,  width: 350,  height: 35);
        tf3 = new JTextField();
        tf3.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf3.setFont(h2);
        tf3.setBounds( x: 290,  y: 200,  width: 310,  height: 35);
        l4 = new JLabel( text: "Blood Group : ");
        l4.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l4.setFont(h1);
        l4.setBounds( x: 715,  y: 195,  width: 350,  height: 35);
        String country6[] = { "Choose your Blood Group", "A+", "O+", "B+", "AB+", "A-", "O-", "B-", "AB-" };
        c5 = new JComboBox(country6);
        c5.setFont(h2);
        c5.setBackground(new Color( r: 220,  g: 220,  b: 250));
        c5.setBounds( x: 900,  y: 195,  width: 310,  height: 35);
        l5 = new JLabel( text: "Department Name : ");
        l5.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l5.setFont(h1);
        l5.setBounds( x: 38,  y: 275,  width: 350,  height: 35);
        String country1[] = { "Choose your Department", "CE", "CSE", "EE", "ECE", "IT", "ME" };
```

```
        c2 = new JComboBox(country1);
        c2.setFont(h2);
        c2.setBackground(new Color( r: 220,  g: 220,  b: 250));
        c2.setBounds( x: 290,  y: 275,  width: 310,  height: 35);
        l6 = new JLabel( text: "Gender : ");
        l6.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l6.setFont(h1);
        l6.setBounds( x: 160,  y: 345,  width: 350,  height: 35);
        String country[] = { "Choose Gender", "Male", "Female", "Others" };
        c1 = new JComboBox(country);
        c1.setFont(h2);
        c1.setBackground(new Color( r: 220,  g: 220,  b: 250));
        c1.setBounds( x: 290,  y: 345,  width: 310,  height: 35);
        l7 = new JLabel( text: "Studing Standard : ");
        l7.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l7.setFont(h1);
        l7.setBounds( x: 660,  y: 268,  width: 350,  height: 35);
        String country2[] = { "Choose your Year", "1st", "2nd", "3rd", "4th" };
        c3 = new JComboBox(country2);
        c3.setFont(h2);
        c3.setBackground(new Color( r: 220,  g: 220,  b: 250));
        c3.setBounds( x: 900,  y: 268,  width: 310,  height: 35);
        l8 = new JLabel( text: "Date Of Birth : ");
        l8.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l8.setFont(h1);
        l8.setBounds( x: 710,  y: 340,  width: 350,  height: 35);
        String country3[] = { "DD", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14",
                "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" };
        cd4 = new JComboBox(country3);
        cd4.setFont(h2);
        cd4.setBackground(new Color( r: 220,  g: 220,  b: 250));
        cd4.setBounds( x: 900,  y: 340,  width: 90,  height: 35);
        String country4[] = { "MM", "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov",
                "Dec" };
        cm4 = new JComboBox(country4);
        cm4.setBackground(new Color( r: 220,  g: 220,  b: 250));
        cm4.setFont(h2);
        cm4.setBounds( x: 990,  y: 340,  width: 100,  height: 35);
        String country5[] = { "YYYY", "1995", "1996", "1997", "1998", "1999", "2000", "2001", "2002", "2003", "2004",
                "2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017",
                "2018", "2019", "2020", "2021", "2022", "2023", "2024", "2025", "2026", "2027", "2028", "2029",
                "2030" };
        cy4 = new JComboBox(country5);
        cy4.setFont(h2);
        cy4.setBackground(new Color( r: 220,  g: 220,  b: 250));
        cy4.setBounds( x: 1090,  y: 340,  width: 120,  height: 35);
        l9 = new JLabel( text: "Email ID : ");
        l9.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l9.setFont(h1);
        l9.setBounds( x: 765,  y: 410,  width: 350,  height: 35);
        tf8 = new JTextField();
        tf8.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf8.setFont(h2);
        tf8.setBounds( x: 900,  y: 410,  width: 310,  height: 35);
        l10 = new JLabel( text: "Phone Number : ");
        l10.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l10.setFont(h1);
        l10.setBounds( x: 70,  y: 410,  width: 350,  height: 35);
        tf9 = new JTextField();
        tf9.setBackground(new Color( r: 220,  g: 220,  b: 250));
        tf9.setFont(h2);
        tf9.setBounds( x: 290,  y: 410,  width: 310,  height: 35);
        l11 = new JLabel( text: "Postal Address : ");
        l11.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l11.setFont(h1);
        l11.setBounds( x: 70,  y: 505,  width: 210,  height: 35);
        ta1 = new TextArea();
        ta1.setBackground(new Color( r: 220,  g: 220,  b: 250));
        ta1.setFont(h2);
        ta1.setBounds( x: 290,  y: 480,  width: 310,  height: 100);
        l12 = new JLabel( text: "Date of Addmission : ");
        l12.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l12.setFont(h1);
        l12.setBounds( x: 25,  y: 615,  width: 350,  height: 35);
        String[] country7 = { "DD", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14",
                "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" };
        cd7 = new JComboBox(country7);
        cd7.setFont(h2);
        cd7.setBackground(new Color( r: 220,  g: 220,  b: 250));
        cd7.setBounds( x: 290,  y: 615,  width: 90,  height: 35);
        String[] country8 = { "MM", "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov",
                "Dec" };
        cm7 = new JComboBox(country8);
        cm7.setFont(h2);
        cm7.setBackground(new Color( r: 220,  g: 220,  b: 250));
        cm7.setBounds( x: 380,  y: 615,  width: 100,  height: 35);
        String[] country9 = { "YYYY", "1995", "1996", "1997", "1998", "1999", "2000", "2001", "2002", "2003", "2004",
                "2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017",
                "2018", "2019", "2020", "2021", "2022", "2023", "2024", "2025", "2026", "2027", "2028", "2029",
                "2030" };
        cy7 = new JComboBox(country9);
        cy7.setFont(h2);
        cy7.setBackground(new Color( r: 220,  g: 220,  b: 250));
        cy7.setBounds( x: 480,  y: 615,  width: 120,  height: 35);
        l13 = new JLabel( text: "Year of Passout : ");
        l13.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l13.setFont(h1);
        l13.setBounds( x: 675,  y: 485,  width: 350,  height: 35);
        String[] country10 = { "YYYY", "1995", "1996", "1997", "1998", "1999", "2000", "2001", "2002", "2003", "2004",
                "2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017",
                "2018", "2019", "2020", "2021", "2022", "2023", "2024", "2025", "2026", "2027", "2028", "2029",
                "2030" };
        c6 = new JComboBox(country10);
        c6.setFont(h2);
        c6.setBackground(new Color( r: 220,  g: 220,  b: 250));
        c6.setBounds( x: 900,  y: 485,  width: 310,  height: 35);
        l14 = new JLabel( text: "Religion : ");
        l14.setForeground(new Color( r: 153,  g: 255,  b: 255));
        l14.setFont(h1);
        l14.setBounds( x: 770,  y: 560,  width: 350,  height: 35);
        String[] country11 = { "Choose Your Religion", "Hinduism", "Islam", "Christianity", "Buddhism", "Others" };
        c8 = new JComboBox(country11);
        c8.setFont(h2);
```

```
        c8.setBackground(new Color( r: 220,  g: 220,  b: 250));
        c8.setBounds( x: 900,  y: 560,  width: 310,  height: 35);
        b1 = new JButton( text: "UPDATE ");
        b1.setBounds( x: 900,  y: 640,  width: 100,  height: 35);
        b2 = new JButton( text: "HOME");
        b2.setBounds( x: 1113,  y: 640,  width: 100,  height: 35);
        f1.add(l1);
        f1.add(l0);
        f1.add(l00);
        f1.add(l1);
        f1.add(tf1);
        f1.add(l2);
        f1.add(tf2);
        f1.add(l3);
        f1.add(tf3);
        f1.add(l4);
        f1.add(c5);
        f1.add(l5);
        f1.add(c2);
        f1.add(c1);
        f1.add(l6);
        f1.add(c1);
        f1.add(l7);
        f1.add(c3);
        f1.add(l8);
        f1.add(cd4);
        f1.add(cm4);
        f1.add(cy4);
        f1.add(l9);
        f1.add(tf8);
        f1.add(l10);
        f1.add(tf9);
        f1.add(l11);
        f1.add(ta1);
        f1.add(l12);
        f1.add(cd7);
        f1.add(cm7);
        f1.add(cy7);
        f1.add(l13);
        f1.add(c6);
        f1.add(l14);
        f1.add(c8);
        f1.add(b1);
        f1.add(b2);
        f1.getContentPane().setBackground(new Color( r: 100,  g: 00,  b: 250));
        f1.setSize( width: 1365,  height: 770);
        f1.setLayout(null);
        f1.setVisible(false);
        f1.setResizable(false);
        f1.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        b0.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e0) {
                    n = Integer.parseInt(String.valueOf(c0.getSelectedItem()));
                    System.out.println(n);
                    f0.setVisible(false);
                    Updation(n);
                    f1.setVisible(true);
                }
        });
        b1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e1) {
                    submitUpdatedDetails(n);
                    f1.dispose();
                    f0.setVisible(true);
                }
        });
        b2.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                f1.dispose();
                new MenuFrame(adminName);
            }
        });
    }
    protected void Updation(int n) {
        System.out.println("================================================================");
        DBConnector db = new DBConnector();
        Connection con = db.connect();
        String sql0 = "SELECT StudentName FROM `student` WHERE StudentID = " + n;
        try {
            Statement st0 = con.createStatement();
            ResultSet rs0 = st0.executeQuery(sql0);
            if (rs0.next()) {
                System.out.println("Student Name : " + rs0.getString( columnLabel: "StudentName"));
                tf1.setText(String.valueOf(n));
                tf2.setText(rs0.getString( columnLabel: "StudentName"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        String sql1 = "SELECT StudentGender FROM `student` WHERE StudentID = " + n;
        try {
            Statement st1 = con.createStatement();
            ResultSet rs1 = st1.executeQuery(sql1);
            if (rs1.next()) {
                System.out.println("Student Gender : " + rs1.getString( columnLabel: "StudentGender"));
                c1.setSelectedItem(rs1.getString( columnLabel: "StudentGender"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        String sql2 = "SELECT BloodGrp FROM `student` WHERE StudentID = " + n;
        try {
            Statement st2 = con.createStatement();
            ResultSet rs2 = st2.executeQuery(sql2);
            if (rs2.next()) {
                System.out.println("Student Blood Group : " + rs2.getString( columnLabel: "BloodGrp"));
```

```
                c5.setSelectedItem(rs2.getString( columnLabel: "BloodGrp"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        String sql3 = "SELECT DepartmentId FROM `student` WHERE StudentID = " + n;
        try {
            Statement st3 = con.createStatement();
            ResultSet rs3 = st3.executeQuery(sql3);
            if (rs3.next()) {
                System.out.println("Student Department Id : " + rs3.getString( columnLabel: "DepartmentId"));
                c2.setSelectedItem(rs3.getString( columnLabel: "DepartmentId"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        String sql4 = "SELECT DOB FROM `student` WHERE StudentID = " + n;
        try {
            Statement st4 = con.createStatement();
            ResultSet rs4 = st4.executeQuery(sql4);
            if (rs4.next()) {
                System.out.println("Student Date Of Birth : " + rs4.getString( columnLabel: "DOB"));
                String str = rs4.getString( columnLabel: "DOB");
                String as[] = str.split( regex: "/");
                cd4.setSelectedItem(String.valueOf(as[0]));
                cm4.setSelectedItem(String.valueOf(as[1]));
                cy4.setSelectedItem(String.valueOf(as[2]));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        String sql5 = "SELECT StudyStandard FROM `student` WHERE StudentID = " + n;
        try {
            Statement st5 = con.createStatement();
            ResultSet rs5 = st5.executeQuery(sql5);
            if (rs5.next()) {
                System.out.println("Student Study Standard : " + rs5.getString( columnLabel: "StudyStandard"));
                c3.setSelectedItem(rs5.getString( columnLabel: "StudyStandard"));
            }
        } catch (SQLException e) { e.printStackTrace(); }
        String sql6 = "SELECT GuardianName FROM `student` WHERE StudentID = " + n;
        try {
            Statement st6 = con.createStatement();
            ResultSet rs6 = st6.executeQuery(sql6);
            if (rs6.next()) {
                System.out.println("Guardian's Name : " + rs6.getString( columnLabel: "GuardianName"));
                tf3.setText(rs6.getString( columnLabel: "GuardianName"));
            }
        } catch (SQLException e) { e.printStackTrace(); }
        String sql7 = "SELECT PhoneNumber FROM `student` WHERE StudentID = " + n;
        try {
            Statement st7 = con.createStatement();
            ResultSet rs7 = st7.executeQuery(sql7);
            if (rs7.next()) {
                System.out.println("Student's Phone Number : " + rs7.getString( columnLabel: "PhoneNumber"));
                tf9.setText(rs7.getString( columnLabel: "PhoneNumber"));
            }
        } catch (SQLException e) { e.printStackTrace(); }
        String sql8 = "SELECT Religion FROM `student` WHERE StudentID = " + n;
        try {
            Statement st8 = con.createStatement();
            ResultSet rs8 = st8.executeQuery(sql8);
            if (rs8.next()) {
                System.out.println("Student's Religion : " + rs8.getString( columnLabel: "Religion"));
                c8.setSelectedItem(rs8.getString( columnLabel: "Religion"));
            }
        } catch (SQLException e) { e.printStackTrace(); }
        String sql9 = "SELECT PostalAddress FROM `student` WHERE StudentID = " + n;
        try {
            Statement st9 = con.createStatement();
            ResultSet rs9 = st9.executeQuery(sql9);
            if (rs9.next()) {
                System.out.println("Student's Postal Address : " + rs9.getString( columnLabel: "PostalAddress"));
                ta1.setText(rs9.getString( columnLabel: "PostalAddress"));
            }
        } catch (SQLException e) { e.printStackTrace(); }
        String sql10 = "SELECT EmailAddress FROM `student` WHERE StudentID = " + n;
        try {
            Statement st10 = con.createStatement();
            ResultSet rs10 = st10.executeQuery(sql10);
            if (rs10.next()) {
                System.out.println("Student's Email Address : " + rs10.getString( columnLabel: "EmailAddress"));
                tf8.setText(rs10.getString( columnLabel: "EmailAddress"));
            }
        } catch (SQLException e) { e.printStackTrace(); }
        String sql11 = "SELECT DateOfAdmission FROM `student` WHERE StudentID = " + n;
        try {
            Statement st11 = con.createStatement();
            ResultSet rs11 = st11.executeQuery(sql11);
            if (rs11.next()) {
                System.out.println("Student's Date Of Admission : " + rs11.getString( columnLabel: "DateOfAdmission"));
                String str1 = rs11.getString( columnLabel: "DateOfAdmission");
                String as1[] = str1.split( regex: "/");
                cd7.setSelectedItem(String.valueOf(as1[0]));
                cm7.setSelectedItem(String.valueOf(as1[1]));
                cy7.setSelectedItem(String.valueOf(as1[2]));
            }
        } catch (SQLException e) { e.printStackTrace(); }
        String sql12 = "SELECT YearOfPassOut FROM `student` WHERE StudentID = " + n;
        try {
            Statement st12 = con.createStatement();
            ResultSet rs12 = st12.executeQuery(sql12);
            if (rs12.next()) {
                System.out.println("Student's Year Of PassOut : " + rs12.getString( columnLabel: "YearOfPassOut"));
                c6.setSelectedItem(String.valueOf(rs12.getInt( columnLabel: "YearOfPassOut")));
            }
        } catch (SQLException e) { e.printStackTrace(); }
    }
```

```
protected void submitUpdatedDetails(int n) {
    DBConnector con = new DBConnector();
    Connection cc = con.connect();
    String SQL = "UPDATE `student` SET `StudentName`='" + String.valueOf(tf2.getText()) + "',`StudentGender`='" + String.valueOf(c1.getItemAt(c1.getSelectedIndex()))
            + "',`BloodGrp`='" + String.valueOf(c5.getItemAt(c5.getSelectedIndex())) + "',`DepartmentID`='" + String.valueOf(c2.getItemAt(c2.getSelectedIndex())) +
            "',`StudyStandard`='" + String.valueOf(c3.getItemAt(c3.getSelectedIndex())) + "',`DOB`='" + String.valueOf(cd4.getSelectedItem() + "/" + cm4.getSelectedItem()
            + "/" + cy4.getSelectedItem()) + "',`GuardianName`='" + String.valueOf(tf3.getText()) + "',`PhoneNumber`='" + Long.parseLong(tf9.getText()) + "',`PostalAddress`='"
            + String.valueOf(ta1.getText()) + "',`EmailAddress`='" + String.valueOf(tf8.getText()) + "',`Religion`='" + String.valueOf(c8.getItemAt(c8.getSelectedIndex())) +
            "',`DateOfAdmission`='" + String.valueOf(cd7.getSelectedItem() + "/" + cm7.getSelectedItem() + "/" + cy7.getSelectedItem()) + "',`YearOfPassOut`= " +
            Integer.parseInt(String.valueOf(c6.getItemAt(c6.getSelectedIndex()))) + " WHERE `StudentID` = " + n ;
    System.out.println(SQL);
    PreparedStatement ps;
    try {
        ps = cc.prepareStatement(SQL);
        int rs = ps.executeUpdate();
        if (rs > 0) {
            JOptionPane.showMessageDialog( parentComponent: null,  message: "Student has been Updated");
        } else {
            JOptionPane.showMessageDialog( parentComponent: null,  message: "Something went wrong, try again");
        }
    } catch (SQLException e) { e.printStackTrace(); }
}
}
```

Update Exam:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class UpdateExamFrame {
    Font font1 = new Font(Font.MONOSPACED,Font.BOLD, size: 25);
    Font font2 = new Font(Font.SANS_SERIF,Font.BOLD, size: 18);
    Font font3 = new Font(Font.DIALOG_INPUT,Font.BOLD, size: 15);
    JFrame frame3 = new JFrame();
    private JLabel updateExamDate, dateLabel,departmentLabel;
    private JTextField dateField;
    private JComboBox departmentBox;
    private JButton submit,returnMenu;
    String[] Option = {"Select","CE","CSE","EE","ECE","IT","ME"};
    JLabel showAdmin;
    String name,department,edate;
    public UpdateExamFrame(String adminName) {
        showAdmin = new JLabel( text: "YOU LOGGED IN AS : " + adminName);
        name = adminName;
        updateExamDate = new JLabel( text: "Update Exam Date");
        departmentLabel = new JLabel( text: "Department Name :");
        departmentBox = new JComboBox(Option);
        dateLabel = new JLabel( text: "Exam Date :");
        dateField = new JTextField();
        submit = new JButton( text: "UPDATE");
        returnMenu = new JButton( text: "HOME");
        showAdmin.setFont(font1);
        showAdmin.setForeground(Color.yellow);
        showAdmin.setBounds( x: 10, y: 10, width: 700, height: 40);
        frame3.add(showAdmin);
        frame3.add(updateExamDate);
        frame3.add(departmentLabel);
        frame3.add(departmentBox);
        frame3.add(dateLabel);
        frame3.add(dateField);
        frame3.add(submit);
        frame3.add(returnMenu);
        updateExamDate.setBounds( x: 60, y: 50, width: 200, height: 100);
        departmentLabel.setBounds( x: 60, y: 130, width: 150, height: 30);
        departmentBox.setBounds( x: 230, y: 130, width: 150, height: 30);
        dateLabel.setBounds( x: 60, y: 170, width: 150, height: 30);
        dateField.setBounds( x: 230, y: 170, width: 150, height: 30);
        submit.setBounds( x: 50, y: 230, width: 150, height: 30);
        returnMenu.setBounds( x: 250, y: 230, width: 150, height: 30);
        updateExamDate.setForeground(new Color( r: 250, g: 250, b: 250));
        departmentLabel.setForeground(new Color( r: 250, g: 250, b: 250));
        dateLabel.setForeground(new Color( r: 250, g: 250, b: 250));
        departmentBox.setBackground(new Color( r: 220,  g: 220,  b: 250));
        updateExamDate.setFont(font2);
        departmentLabel.setFont(font3);
        dateLabel.setFont(font3);
        returnMenu.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame3.dispose();
                new MenuFrame(adminName);
            }
        });
```

```
        submit.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                department = String.valueOf(departmentBox.getItemAt(departmentBox.getSelectedIndex()));
                edate = String.valueOf(dateField.getText());
                insertion();
                new MenuFrame(adminName);
            }
        });
        frame3.setSize( width: 500, height: 350); //400 width and 500 height
        frame3.getContentPane().setBackground(new Color( r: 100, g: 80, b: 250));
        frame3.setLayout(null); //using no layout managers
        frame3.setVisible(true); //making the frame visible
        frame3.setResizable(false);
        frame3.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    }
    protected void insertion() {
        DBConnector con = new DBConnector();
        Connection cc = con.connect();
        String sql = "UPDATE `examination` SET `DateOfExamination`= ? WHERE `DepartmentID`= ?";
        PreparedStatement ps;
        try {
            ps = cc.prepareStatement(sql);
            ps.setString( parameterIndex: 1, edate);
            ps.setString( parameterIndex: 2, department);
            int rs = ps.executeUpdate();
            if(rs >0) {
                JOptionPane.showMessageDialog( parentComponent: null,  message: "Exam Date Updated");
                frame3.dispose();
            }
            else {
                JOptionPane.showMessageDialog( parentComponent: null,  message: "Try again");
            }
        }
        catch (SQLException e) { e.printStackTrace(); }
    }
}
```

Display Student:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class DisplayStudentFrame {
    Font font1 = new Font(Font.MONOSPACED,Font.BOLD, size: 20);
    Font font2 = new Font(Font.SANS_SERIF,Font.BOLD, size: 18);
    Font font3 = new Font(Font.DIALOG_INPUT,Font.BOLD, size: 50);
    Font font4 = new Font(Font.SANS_SERIF,Font.BOLD, size: 9);
    JFrame frame6 = new JFrame();
    JTable table;
    private JComboBox departmentBox;
    private JButton ok,homeButton;
    private JLabel departmentLabel;
    JLabel showAdmin;
    String name;
    String[] Option = {"Select","CE","CSE","EE","ECE","IT","ME"};
    public DisplayStudentFrame(String adminName){
        showAdmin = new JLabel( text: "YOU LOGGED IN AS : " + adminName);
        name = adminName;
        departmentLabel = new JLabel( text: "Department Name :");
        departmentBox = new JComboBox(Option);
        ok = new JButton( text: "SUBMIT");
        homeButton = new JButton( text: "HOME");
        showAdmin.setFont(font1);
        showAdmin.setForeground(Color.yellow);
        showAdmin.setBounds( x: 10, y: 10, width: 700, height: 40);
        frame6.add(showAdmin);
        frame6.add(departmentLabel);
        frame6.add(departmentBox);
        frame6.add(ok);
        frame6.add(homeButton);
        departmentLabel.setForeground(new Color( r: 255,  g: 255,  b: 255));
        departmentLabel.setFont(font2);
        departmentBox.setBackground(new Color( r: 220,  g: 220,  b: 250));
        departmentLabel.setBounds( x: 60, y: 120, width: 200, height: 50);
        departmentBox.setBounds( x: 265, y: 130, width: 150, height: 30);
        ok.setBounds( x: 60, y: 230, width: 150, height: 30);
        homeButton.setBounds( x: 265, y: 230, width: 150, height: 30);
        homeButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame6.dispose();
                new MenuFrame(adminName);
            }
        });
        ok.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String deptName =  String.valueOf(departmentBox.getItemAt( departmentBox.getSelectedIndex()));
                DisplayTable(deptName);
            }
        });
        frame6.setSize( width: 500, height: 350); //400 width and 500 height
        frame6.getContentPane().setBackground(new Color( r: 100, g: 80, b: 250));
        frame6.setLayout(null); //using no layout managers
        frame6.setVisible(true); //making the frame visible
        frame6.setResizable(false);
        frame6.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    }
    private void DisplayTable(String deptName)
    {
        JFrame frame7 = new JFrame();
        JLabel frameLabel_1 = new JLabel( text: "STUDENT LIST");
        JLabel frameLabel_2 = new JLabel( text: "STUDENT LIST");
        frameLabel_1.setFont(font3);
        frameLabel_2.setFont(font3);
```

```
        frameLabel_1.setForeground(new Color( r: 255,  g: 166,  b: 0));
        frameLabel_2.setForeground(new Color( r: 248,  g: 238,  b: 31));
        frameLabel_1.setBounds( x: 550, y: 0, width: 500, height: 145);
        frameLabel_2.setBounds( x: 552, y: 2, width: 500, height: 145);
        frame7.add(frameLabel_1);
        frame7.add(frameLabel_2);
        DefaultTableModel tableModel = new DefaultTableModel();
        table = new JTable(tableModel);
        tableModel.addColumn(  columnName: "Student ID" );
        tableModel.addColumn(  columnName: "Student Name" );
        tableModel.addColumn(  columnName: "Gender" );
        tableModel.addColumn(  columnName: "Blood Grp" );
        tableModel.addColumn(  columnName: "Department" );
        tableModel.addColumn(  columnName: "Study Standard" );
        tableModel.addColumn(  columnName: "Date of Birth" );
        tableModel.addColumn(  columnName: "Guardian Name" );
        tableModel.addColumn(  columnName: "Phone Number" );
        tableModel.addColumn(  columnName: "Postal Address" );
        tableModel.addColumn(  columnName: "Email Address" );
        tableModel.addColumn(  columnName: "Religion" );
        tableModel.addColumn(  columnName: "Date of Admission" );
        tableModel.addColumn(  columnName: "Year of Passout" );
        JScrollPane jScrollPane = new JScrollPane(table);
        jScrollPane.setBounds( x: 1,  y: 150,  width: 1530,  height: 2000);
        table.setBackground(Color.cyan);
        table.setFont(font4);
        frame7.add(jScrollPane);
        frame7.setSize( width: 1590, height: 500); //400 width and 500 height
        frame7.getContentPane().setBackground(new Color( r: 107,  g: 0,  b: 255));
        frame7.setLayout(null); //using no layout managers
        frame7.setVisible(true); //making the frame visible
        frame7.setResizable(true);
        frame7.addWindowListener(WindowAdapter) windowClosed(e) -> { frame7.dispose(); });
        DBConnector con = new DBConnector();
        Connection cc = con.connect();
        String str = "select * from student where departmentID='" + deptName + "'";
          PreparedStatement ps;
          try {
              ps = cc.prepareStatement(str);
              ResultSet rs =ps.executeQuery(str);
              System.out.println(rs);
              table.setModel(DbUtils.resultSetToTableModel(rs));
          }catch( Exception e) {
              e.printStackTrace();
          }
          table.getColumnModel().getColumn( columnIndex: 0).setPreferredWidth(1);
          table.getColumnModel().getColumn( columnIndex: 1).setPreferredWidth(8);
          table.getColumnModel().getColumn( columnIndex: 2).setPreferredWidth(5);
          table.getColumnModel().getColumn( columnIndex: 3).setPreferredWidth(1);
          table.getColumnModel().getColumn( columnIndex: 4).setPreferredWidth(1);
          table.getColumnModel().getColumn( columnIndex: 5).setPreferredWidth(1);
          table.getColumnModel().getColumn( columnIndex: 7).setPreferredWidth(8);
          table.getColumnModel().getColumn( columnIndex: 11).setPreferredWidth(1);
          table.getColumnModel().getColumn( columnIndex: 12).setPreferredWidth(1);
          table.getColumnModel().getColumn( columnIndex: 13).setPreferredWidth(1);
          table.getColumnModel().getColumn( columnIndex: 6).setPreferredWidth(4);
          table.getColumnModel().getColumn( columnIndex: 8).setPreferredWidth(4);
    }
}
```

Display Exam:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class DisplayExamFrame {
    Font font1 = new Font(Font.MONOSPACED,Font.BOLD, size: 30);
    Font font2 = new Font(Font.DIALOG_INPUT, Font.BOLD, size: 15);
    JFrame frame4 = new JFrame();
    JTable table;
    private JComboBox departmentBox;
    private JButton homeButton;
    public DisplayExamFrame(String adminName) {
        JLabel frameLabel_1 = new JLabel( text: "EXAM DATE LIST");
        JLabel frameLabel_2 = new JLabel( text: "EXAM DATE LIST");
        DefaultTableModel tableModel = new DefaultTableModel();
        tableModel.addColumn( columnName: "Department");
        tableModel.addColumn( columnName: "Exam Date");
        homeButton = new JButton( text: "Home");
        table = new JTable(tableModel);
        JScrollPane jScrollPane = new JScrollPane(table);
        frameLabel_1.setFont(font1);
        frameLabel_2.setFont(font1);
        table.setFont(font2);
        frameLabel_1.setForeground(new Color( r: 255,  g: 166,  b: 0));
        frameLabel_2.setForeground(new Color( r: 248,  g: 238,  b: 31));
        table.setBackground(Color.cyan);
        frameLabel_1.setBounds( x: 118, y: 1, width: 300, height: 50);
        frameLabel_2.setBounds( x: 120, y: 3, width: 300, height: 50);
        jScrollPane.setBounds( x: 1,  y: 60,  width: 500,  height: 122);
        homeButton.setBounds( x: 150, y: 200, width: 150, height: 30);
```

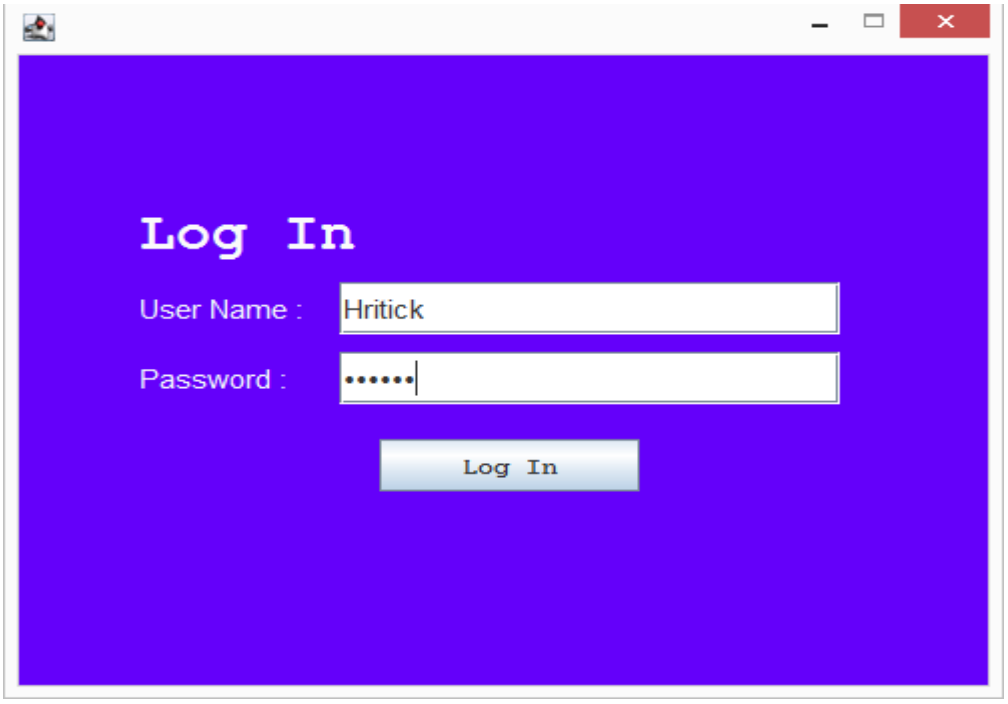
```
        frame4.add(frameLabel_1);
        frame4.add(frameLabel_2);
        frame4.add(jScrollPane);
        frame4.add(homeButton);
        frame4.setSize( width: 500, height: 300); //400 width and 500 height
        frame4.getContentPane().setBackground(new Color( r: 100, g: 0, b: 250));
        frame4.setLayout(null); //using no layout managers
        frame4.setVisible(true); //making the frame visible
        frame4.setResizable(false);
        frame4.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        homeButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new MenuFrame(adminName);
                frame4.dispose();
            }
        });
        DBConnector con = new DBConnector();
        Connection cc = con.connect();
        String str = "select d.departmentName as Department, e.DateOfExamination as 'Exam Date' from examination e," +
                " department d where d.departmentID = e.departmentID order by d.departmentName";
        PreparedStatement ps;
        try {
            ps = cc.prepareStatement(str);
            ResultSet rs =ps.executeQuery(str);
            System.out.println(rs);
            table.setModel(DbUtils.resultSetToTableModel(rs));
            table.getColumnModel().getColumn( columnIndex: 0).setPreferredWidth(250);
        }catch(Exception e) { e.printStackTrace(); }
    }
}
```

DB Connector:

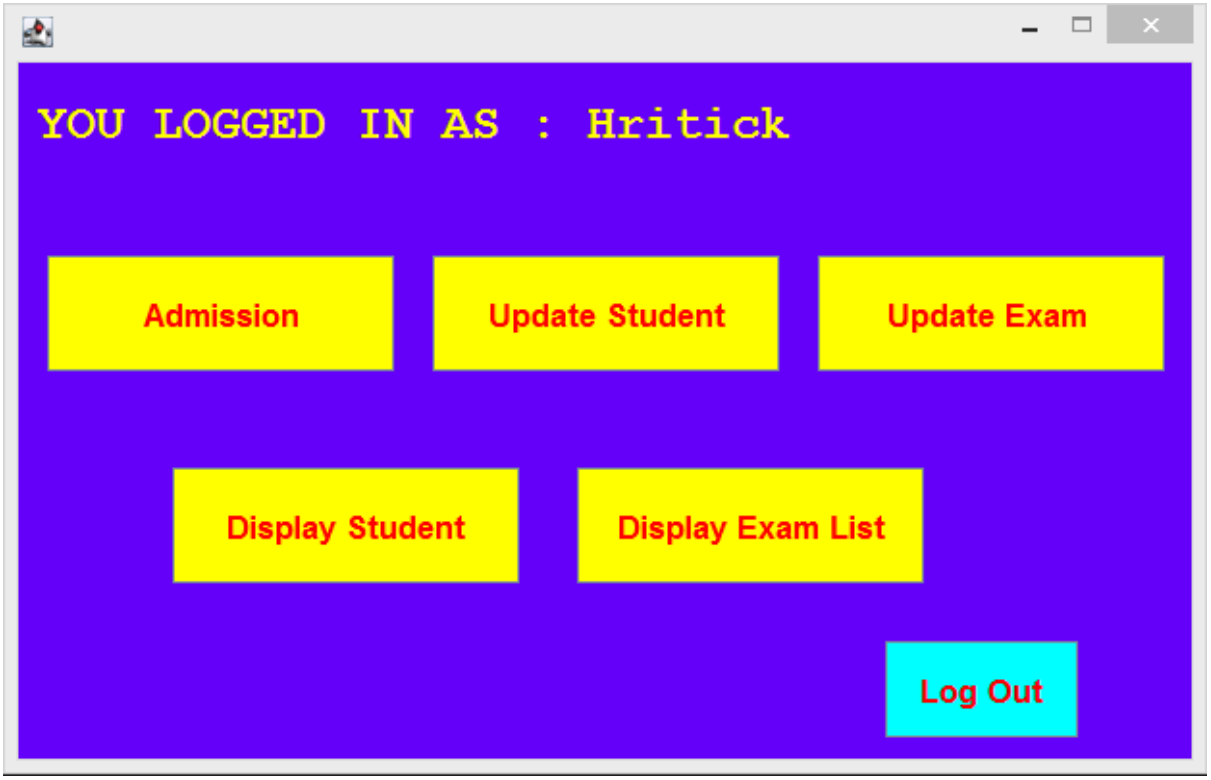
```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class DBConnector {
    private static final String DATABASE_DRIVER = "com.mysql.cj.jdbc.Driver";
    private static final String DATABASE_URL = "jdbc:mysql://localhost:3306/student management system";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "";
    private static final String MAX_POOL = "250";
    // declaring connection object
    private Connection connection;
    // declaring properties object
    private Properties properties;
    // create properties
    private Properties getProperties() {
        if (properties == null) {
            properties = new Properties();
            properties.setProperty("user", USERNAME);
            properties.setProperty("password", PASSWORD);
            properties.setProperty("MaxPooledStatements", MAX_POOL);
        }
        return properties;
    }
    // connect database
    public Connection connect() {
        if (connection == null) {
            try {
                Class.forName(DATABASE_DRIVER);
                connection = DriverManager.getConnection(DATABASE_URL, getProperties());
            } catch (ClassNotFoundException | SQLException e) {
                e.printStackTrace();
            }
        }
        return connection;
    }
    // disconnect database
    public void disconnect() {
        if (connection != null) {
            try {
                connection.close();
                connection = null;
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

}
```

Output Screenshot:-

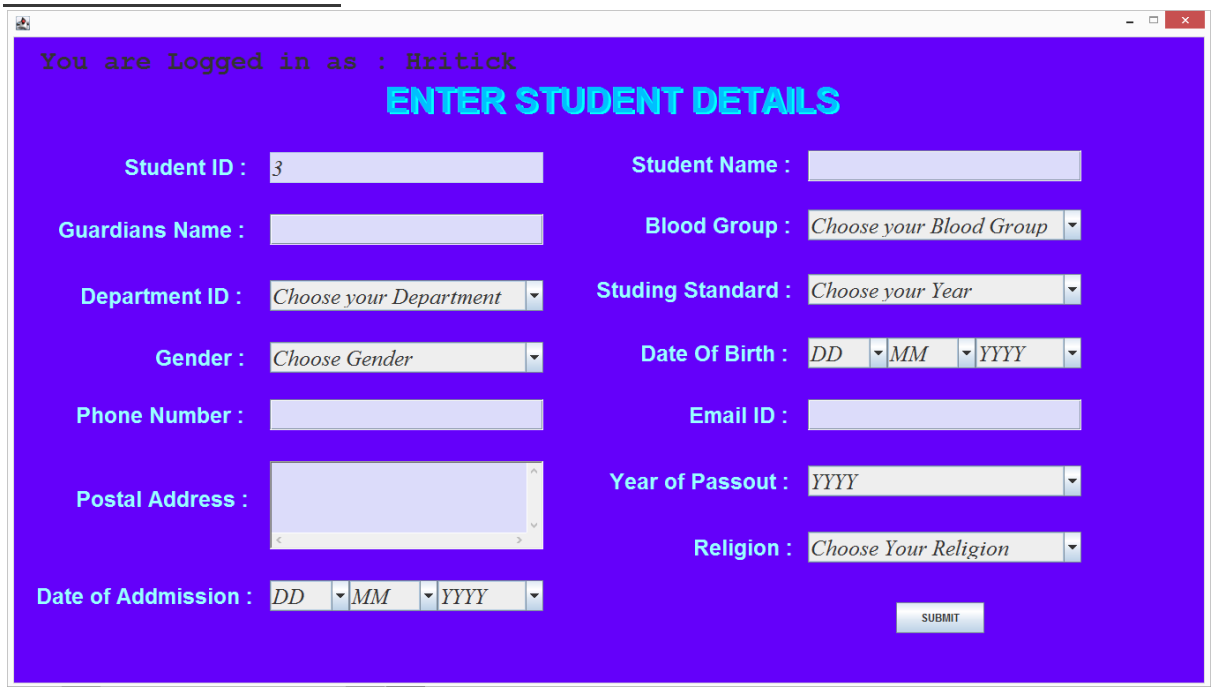
Log in Screen:

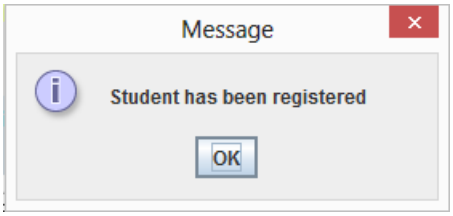


Menu Screen:

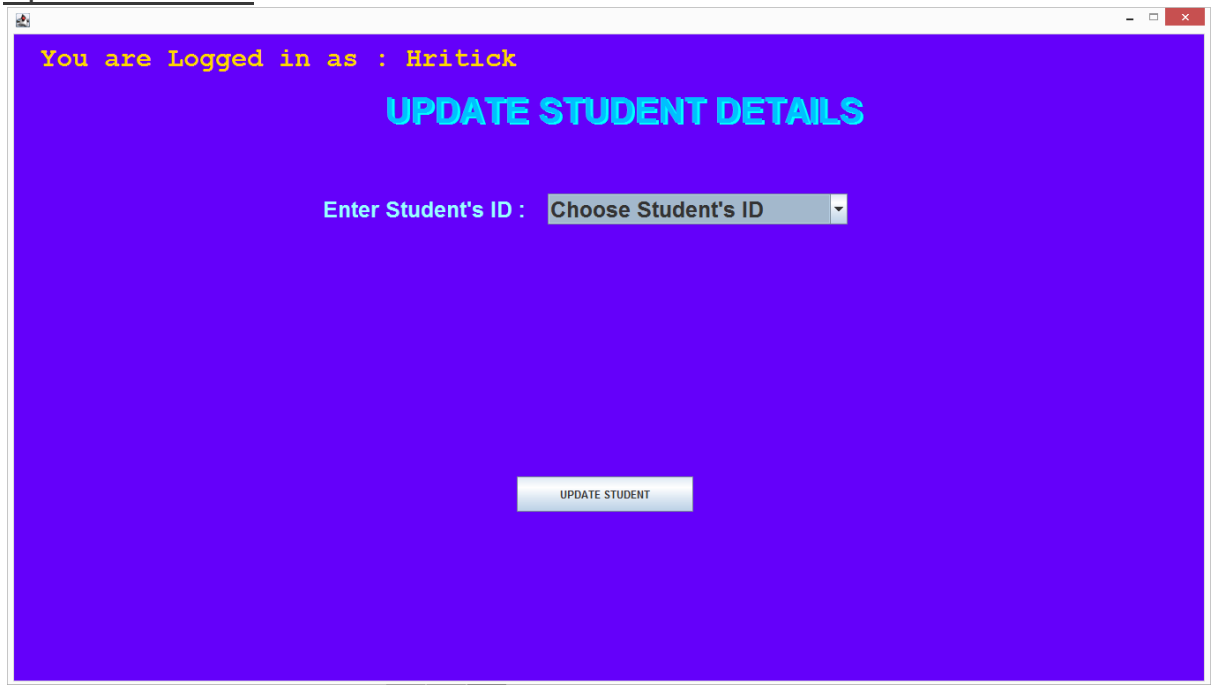


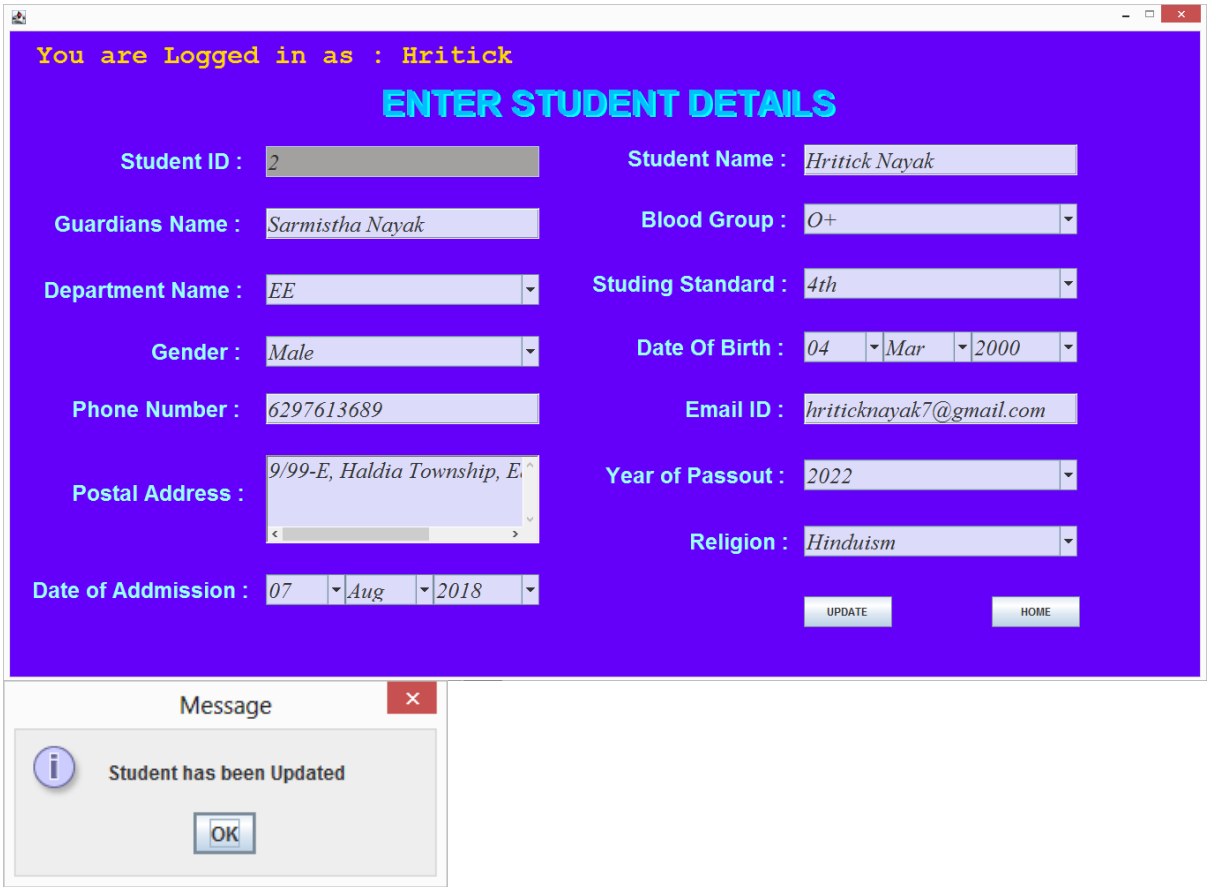
Admission Student:



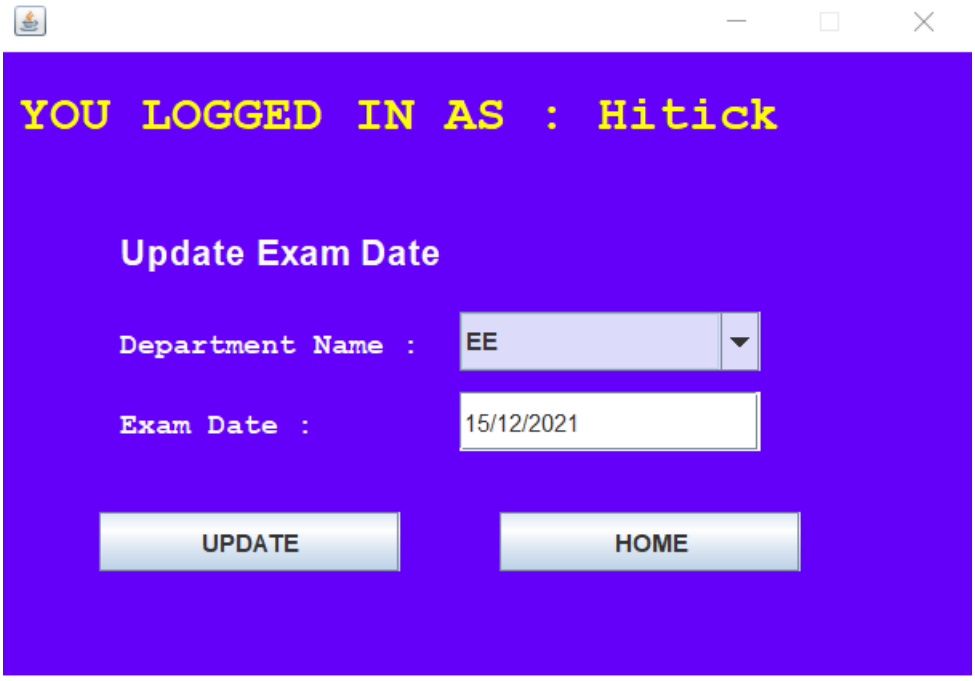


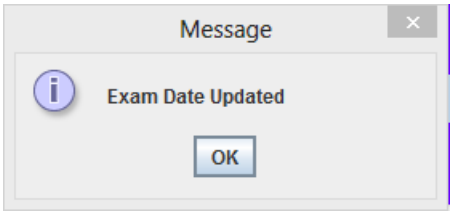
Update Student:



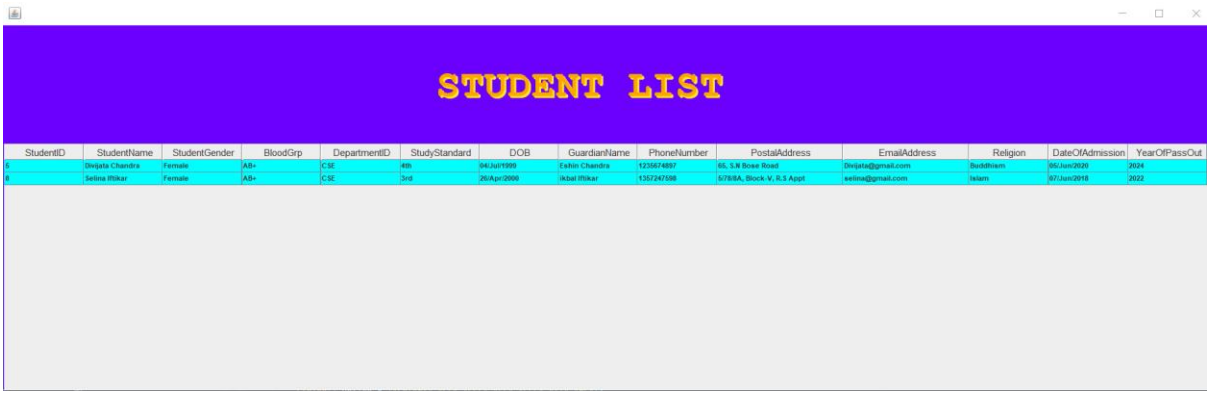


Update Exam Date:





Display Student:



TESTING AND MAINTENANCE

Objective of Testing:-

The major objectives of testing are -

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To make sure that the end result meets the business and user requirements.
- To gain the confidence of the customers by providing them a quality product.

SECURITY MEASURE

Database Security Measure:-

Database Security means to keep sensitive information safe and prevent the loss of data. Security of data base is controlled by Database Administrator. Database security has become an important more than ever just because data breach can be disastrous for any project so below are some steps/precautions project managers can take to keep their sensitive data safe.

- We have to invest in a good antivirus to protect the data so that our files don't get corrupted by any malware.

- We have to use a strong password for our system and be careful about the breach.

- We need to look out for the hardware and backup. Daily backup and hardware checking is required to not risk the losing of data due to hardware crashes.

- Management should be trained to not disclose personal and private information to anyone who is not appropriate to have the information

App security measures:-

- Proper Authentication methods
- Proper Authorization methods
- Proper Encryption of app
- APP security testing on a daily or weekly basis
- Management of granting privilege to the handler

Limitation of App:-

As it's not available online or cloud based so for now user need to go to the available centers to use it.

FUTURE SCOPE

The student management system is a desktop application with a local scope i.e it can be used only on the administrative office. The employees or anyone who needs to operate it has to be present physically there. But the main future scope idea is as of now the availability of the software and data related to it on an online based system. The includes the usage of cloud connected to the application and required dedicated server with proper maintenance, which will enable the admin and other workers to access the system from wherever they want and their work don't get restricted due to physical mode of hindrance to reach the center. And in future it can also be developed for mobile.

CONCLUSION

The project student management system is for computerizing the working in administrative office. This software takes care of all the requirements of administrative office and is capable to provide easy and effective storage of information related to students. It gives access to manage all the details of a student.

It is easy to use, since it uses the GUI provided in the user dialog. User friendly screen are provided. The usage of software increases the efficiency, decreases the effort. It has been thoroughly tested and implemented.

REFERENCES

- ARDENT TRAINING VIDEOS
- https://www.geeksforgeeks.org
- https://www.tutorialspoint.com
- https://www.w3schools.com
- https://www.javatpoint.com