STUDENT RECORDS MANAGEMENT

MINI PROJECT REPORT

Submitted by

ABINAYASHREE S V(953622104001)

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAMCO INSTITUTE OF TECHNOLOGY,

RAJAPALAYAM

JUNE 2024

CS3481 DATABASE MANAGEMENT SYSTEMS

BONAFIDE CERTIFICATE

Certified that this project report titled **“STUDENT RECORD MANAGEMENT”** is the bonafide work of **S.V.ABINAYASHREE (953622104001), G.AISHWARYA (953622104003) and A.ANNALAKSHMI (953622104005)** who carried out the project work under my supervision.

SIGNATURE
Dr.K.Vijayalakshmi, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor & Head
Department of Computer Science and Engineering
Ramco Institute of Technology North Venganallur Village
Rajapalayam-626117.

SIGNATURE
Mrs.K.Jeyakarthika, M.Tech.,
SUPERVISOR
Assistant Professor
Department of Computer Science and Engineering
Ramco Institute of Technology
North Venganallur Village
Rajapalayam-626117.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

This project aims to develop a robust Student Record Management System (SRMS) designed to streamline the management of student data within educational institutions. Leveraging a database-driven approach, the SRMS utilizes MySQL as the backend database for its reliability and scalability. Java, integrated with Spring Boot, forms the core of the application logic, ensuring efficient data processing and seamless integration of system components.

The frontend interface is developed using HTML, CSS, and JavaScript, providing a responsive and user-friendly experience across various devices. Key achievements include significant improvements in data retrieval efficiency, strengthened data integrity through centralized management, enhanced user experience for students, faculty, and administrators alike, and streamlined administrative processes.

Future enhancements are envisioned to include mobile application integration forenhanced accessibility, advanced analytics capabilities to facilitate informed decision-making, integration with learning management systems (LMS) for comprehensive academic management, enhanced security measures to safeguard sensitive student data, and scalability features to accommodate growing institutional needs.

Overall, the SRMS aims to revolutionize student record management by optimizing operational processes, enhancing data-driven insights, and improving overall administrative efficiency within educational institutions

CHAPTER 1
INTRODUCTION

1.1 STUDENT RECORDS MANAGEMENT

Efficient management of student records is critical for educational institutions to uphold accurate academic records and streamline administrative processes effectively. Traditional paper-based systems are inherently prone to errors, data redundancy, and vulnerabilities in terms of security, necessitating a transition towards more reliable digital solutions.

This project proposes the development of a comprehensive Student Record Management System (SRMS) built upon a robust database management system. The SRMS aims to centralize student information, thereby consolidating data from various sources into a unified repository. This consolidation enhances data integrity by minimizing inconsistencies and improving accuracy in record- keeping.

Moreover, the system is designed to facilitate seamless accessibility for authorized users, including students, faculty, and administrators, ensuring they can retrieve and update information efficiently. By embracing a digital platform, educational institutions can not only mitigate the challenges associated with manual processes but also achieve heightened levels of security and compliance with data protection regulations.

The SRMS not only addresses current operational inefficiencies but also lays the groundwork for future advancements in educational management through enhanced analytical capabilities and scalability to support growing institutional needs.

1.2 PROGRAM DESCRIPTION

The Student Record Management System (SRMS) is a robust platform meticulously designed to centralize and streamline the management of student information, course enrollments, and academic records within educational institutions. This system empowers students with secure access to their personal information, allowing them to update details, view grades, and monitor course enrollments conveniently.

Faculty members leverage the SRMS to efficiently submit grades, manage course materials, and foster effective communication with students through integrated messaging features. Administrators benefit from comprehensive user management capabilities, facilitating tasks such as user account creation, role assignment, and permissions management to ensure data security and access control.

Furthermore, administrators can generate in-depth reports on student performance and system usage statistics, providing invaluable insights for informed decision-making and strategic planning within the institution. By enhancing operational efficiency, promoting data accuracy, and fostering seamless communication and collaboration across all levels, the SRMS stands as a cornerstone in modern educational management, poised to elevate administrative capabilities and student success metrics alike

1.3 MODULES

The SRMS comprises several interconnected modules that cater to different user roles and functionalities.

The Student Module:

Student module enables students to securely log in and access their personal details, including contact information and academic records. Students can view their course schedules, grades, and attendance records, as well as update their contact information through the system.

The Faculty Module:

This module allows instructors to manage courses effectively by accessing student records, submitting grades, and distributing course materials. Faculty members can view class rosters, monitor student progress, and communicate withstudents directly through the system.

The Administrator Module:

Here the module empowers administrative staff with the tools to manage user accounts, generate comprehensive reports on student performance, and enforce data security through role-based access control mechanisms. These modules collectively enhance administrative efficiency, promote transparency in academic processes, and ensure secure access to critical student information.

CHAPTER 2

2.SYSTEM SPECIFICATION

2.1Hardware specification

Processor	: Intel dual core
Processor speed:	3.00GHZ
Ram	: 8 GB

- Monitor
- Keyboard
- Mouse

2.2 Software specification

- NetBeans,MYSQL Server

2.3 TECHNOLOGY STACK

The technology stack employed in developing the SRMS includes a robust combination of backend and frontend technologies tailored to meet project requirements effectively. The backend utilizes MySQL as the database management system, chosen for its reliability, scalability, and compatibility withthe project's data management needs.

Java, integrated with the Spring Boot framework, provides the foundation for developing scalable and efficient backend services, ensuring seamless data processing and business logic implementation within the SRMS. On the frontend, HTML, CSS, and JavaScript were employed to create a responsive and user- friendly interface that facilitates intuitive navigation and accessibility across various devices.

The integration of Spring Security enhances data security by enforcing role-based access control mechanisms throughout the system, safeguarding sensitive student information and maintaining compliance with data protection regulations.

Database Management System (DBMS):

MySQL: Chosen for its reliability, scalability, and compatibility.

Backend:

Java: Provides robustness and scalability.

Spring Boot: Framework for rapid application development and integration.

Frontend:

JavaScript: Builds a responsive and user-friendly interface.

Security:

Spring Security: Implements role-based access control (RBAC) and ensuresdata security.

Integration:

Hibernate: Object-relational mapping (ORM) for database interaction.

Deployment: Apache Tomcat: Web server for hosting the application.

2.4 E-R DIAGRAM

An Entity-Relationship (E-R) diagram is a visual representation of the entities (objects or concepts) and their relationships in a system. In the context of a management system, such as an inventory management system, the E-R diagram depicts the structure of the system, including the entities involved and the relationships between them. Here's a general description of an E-R diagram for a management system:

1. Entities:

- The E-R diagram identifies the main entities or objects within the management system. These entities represent the key components or concepts in the system.
- For an inventory management system, common entities may include products, suppliers, customers, orders, sales, shipments, and warehouses.
- Each entity is typically represented by a rectangle in the diagram, with the entity name written inside.

2. Attributes:

- Each entity in the E-R diagram has attributes associated with it. Attributes represent the characteristics or properties of the entities.
- For example, a "Product" entity may have attributes such as product ID, name, description, price, quantity, and supplier ID.
- Attributes are usually listed next to the corresponding entity in the diagram, providing additional details about the entities.

3. Relationships:

- Relationships illustrate how entities are connected or related to each other within the management system.
- Relationships represent associations, dependencies, or interactions between entities.
- Common types of relationships in a management system may include "one-to-many," "many-to-many," or "one-to-one" relationships.
- Relationships are typically depicted as lines connecting the related entities, with labels describing the nature of the relationship.

4. Cardinality and Constraints:

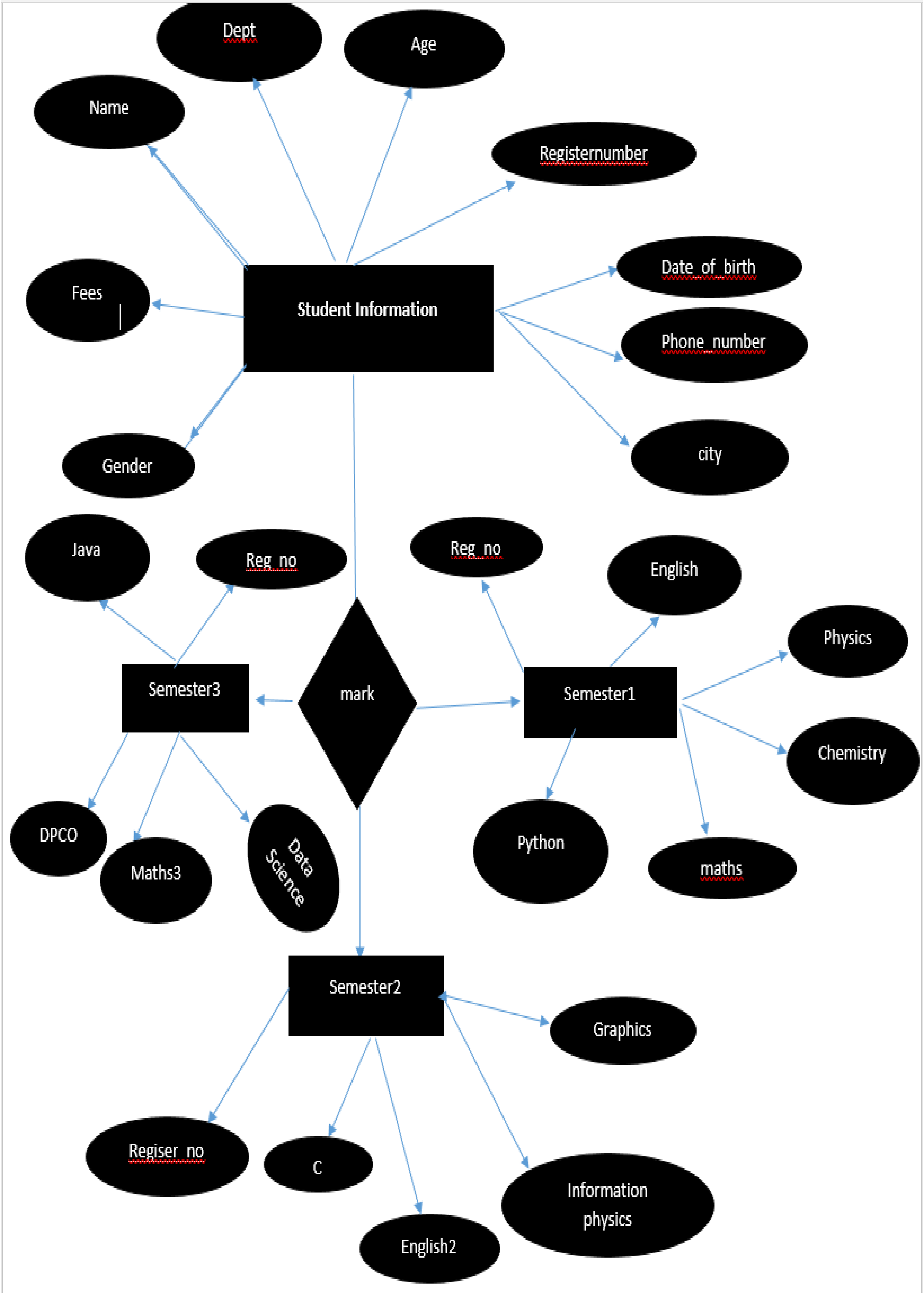
- Cardinality refers to the number of instances of one entity that can be associated with the instances of another entity in a relationship.
- Cardinality constraints, such as "one," "many," or specific numerical limits, define

the allowed associations between entities.

- Cardinality and constraints are often represented using notations like crow's foot, Chen's notation, or UML notation in the E-R diagram.

5. **Primary and Foreign Keys**:
   - Primary keys are unique identifiers assigned to each instance of an entity, ensuring its uniqueness within the system.
   - Foreign keys are attributes that establish relationships between entities by referencing the primary key of another entity.
   - Primary and foreign keys are essential for maintaining data integrity and enabling data retrieval and manipulation.

6. **Additional Diagram Elements:**
   - Depending on the complexity and requirements of the management system, the E-R diagram may include other elements such as weak entities, inheritance hierarchies, or aggregation relationships.
   - These elements help represent specific system characteristics or behaviors that are important for the management system's design.



ER DIAGRAM FOR STUDENT RECORDS MANAGEMENT

CHAPTER 3

3.1 Normalization

First Normal Form (1NF)

Definition:

A table is in 1NF if it only contains atomic (indivisible) values, and each record is unique.

Steps to Achieve 1NF:

1. Eliminate Repeating Groups: Ensure that each column contains only one value per record. For instance, if a student is enrolled in multiple courses, create a separate row for each course the student is enrolled in.
2. Create Separate Columns for Each Attribute: Ensure that each attribute has its own column. Avoid using multi-valued attributes (e.g., a list of courses in one cell).
3. Identify a Primary Key: Ensure that each table has a primary key that uniquely identifies each record.

Second Normal Form (2NF)

Definition:

A table is in 2NF if it is in 1NF and all non-key attributes are fully functionally dependent on the primary key. This means removing partial dependencies where an attribute is dependent on only part of a composite primary key.

Steps to Achieve 2NF:

1. Remove Partial Dependencies: Create separate tables to ensure that non-key attributes are fully dependent on the entire primary key.
2. Create Relationships: Establish relationships between the new tables through foreign keys.

Third Normal Form (3NF)

Definition:

A table is in 3NF if it is in 2NF and all the attributes are only dependent on the primary key, meaning there are no transitive dependencies.

Steps to Achieve 3NF:

1. Remove Transitive Dependencies: Ensure that non-key attributes are not dependent on other non-key attributes. If a non-key attribute depends on another non-key attribute, create a new table to eliminate the transitive dependency.
2. Ensure Direct Dependency on Primary Key: Each non-key attribute should only depend on the primary key

CHAPTER 4

SOURCE CODE

```
package com.mycompany.mini; import
javafx.application.Application;
import static javafx.application.Application.launch;import
javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos; import
javafx.scene.Scene;
import javafx.scene.control.Button; import
javafx.scene.control.DatePicker; import
javafx.scene.control.Label; import
javafx.scene.control.RadioButton;import
javafx.scene.control.TextField;
import javafx.scene.control.ToggleGroup;import
javafx.scene.layout.GridPane; import
javafx.scene.layout.VBox;
import javafx.stage.Stage; import
java.sql.Connection; import
java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement; import
java.util.logging.Level; import
java.util.logging.Logger;
import javafx.scene.control.Alert; import
javafx.scene.control.DialogPane;import
javafx.scene.control.ScrollPane;
public class App extends Application { private
    static final String user = "root";
    private static final String password = "a1b2c3";private
    static final String url =
"jdbc:mysql://localhost:3306/miniproject?zeroDateTimeBehavior=CONVERT_TO_NULL";
    @Override
    public void start(Stage stage) {
        VBox gp1=new VBox();
        Button l1=new Button ("User Login"); Button
        l2=new Button("Admin Login");
        gp1.getChildren().add(l1);
        gp1.getChildren().add(l2);gp1.setSpacing(10);
        gp1.setAlignment(Pos.CENTER); Scene
        sc11=new Scene(gp1,350,200);
        stage.setTitle("Home page");
        stage.setScene(sc11);
       stage.show();
       EventHandler<ActionEvent> stud1 = new EventHandler<ActionEvent>() {public void
```

```
        handle(ActionEvent e) {
        VBox vb1=new VBox();
        Button but=new Button("View The Student Details");
       vb1.getChildren().add(but); vb1.setAlignment(Pos.CENTER);
       Scene sc112=new Scene(vb1,350,200);
        stage.setTitle("Home page");
        stage.setScene(sc112);
      stage.show(); EventHandler<ActionEvent>
       stud123;
            stud123 = new EventHandler<ActionEvent>() {public
                void handle(ActionEvent e) {
                    VBox vbox1 = new VBox(); Button
                    back=new Button("back");
                    vbox1.getChildren().add(back);
                    Scene sc1123=new Scene(vbox1,350,200);
                    stage.setTitle("Home page");
                    stage.setScene(sc1123);
                    stage.show();


                                try(Connection con1 = DriverManager.getConnection(url,
user, password);
                                    PreparedStatement preparedStatement =
con1.prepareStatement("INSERT INTO studentinformation (Register_Number,Name,
Date_Of_Birth, Age, Gender, Phone_Number, City, Fees, Department) VALUES (?, ?, ?, ?,
?, ?, ?, ?, ?)")){
                                Statement statement1 = con1.createStatement();

                                ResultSet resultSet = statement1.executeQuery("SELECT *FROM
studentinformation");

                                if (resultSet.next()) {
                                    EventHandler <ActionEvent> consult1=new
EventHandler<ActionEvent>(){
                                        public void handle(ActionEvent e){
                                            stage.setScene(sc112);
                                        }
                                    };
                                    back.addEventHandler(ActionEvent.ACTION, consult1);
                                    //while (resultSet.next()) { vbox1.getChildren().add(new
                                    Label("STUDENT
DETAILS"));
                                    vbox1.getChildren().add(new Label("Register Number: "
+ (resultSet.getString("Register_Number"))));
                                    vbox1.getChildren().add(new Label("Name: " +
(resultSet.getString("Name"))));
                                    vbox1.getChildren().add(new Label("Date of Birth: "
+(resultSet.getString("Date_Of_Birth")) ));
                                    vbox1.getChildren().add(new Label("Age: "
+(resultSet.getString("Age"))) );
                                    vbox1.getChildren().add(new Label("Gender: " +
(resultSet.getString("Gender"))));
                                    vbox1.getChildren().add(new Label("Phone Number: " +
(resultSet.getString("Phone_Number"))));
                                    vbox1.getChildren().add(new Label("City: "
+(resultSet.getString("City")) ));
                                    vbox1.getChildren().add(new Label("Fees: "
+(resultSet.getString("Fees"))) );
                                    vbox1.getChildren().add(new Label("Departtment: " +
(resultSet.getString("Department"))));

                                           vbox1.getChildren().add(new Label("-----------------------
_______________________________________________"));
                                    // }
                                 }else {


                                    System.out.println("No data found in the database.");
                                }
                            } catch (SQLException ex) {
                                ex.printStackTrace();
                            }
                            try(Connection con1 = DriverManager.getConnection(url,
user, password);
                                    PreparedStatement preparedStatement =
con1.prepareStatement("INSERT INTO semester1(subject_1, grade_1,
subject_2, grade_2, subject_3, grade_3, subject_4, grade_4, subject_5, grade_5)VALUES (?,
?, ?, ?, ?, ?, ?,?,?,?)")){
                                Statement statement1 = con1.createStatement();

                            ResultSet       resultSet = statement1.executeQuery("SELECT *FROM
semester1");
                                if (resultSet.next()) {

                                    EventHandler <ActionEvent> consult1=new
EventHandler<ActionEvent>(){
                                        public void handle(ActionEvent e){
                                            stage.setScene(sc112);
                                        }
                                    };
                                    back.addEventHandler(ActionEvent.ACTION, consult1);
                                   // while (resultSet.next()) {
                                    vbox1.getChildren().add(new Label("SEMESTER -1" ));
                                    vbox1.getChildren().add(new Label("Subject NAME: " +
(resultSet.getString("subject_1"))));
                                    vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_1"))));
                                    vbox1.getChildren().add(new Label("Subject Name: "
+(resultSet.getString("subject_2")) ));
                                    vbox1.getChildren().add(new Label("Grade: "
+(resultSet.getString("grade_2"))) );
                                    vbox1.getChildren().add(new Label("Subject Name: " +
(resultSet.getString("subject_3"))));
                                    vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_3"))));
                                    vbox1.getChildren().add(new Label("Subject Name: "
+(resultSet.getString("subject_4")) ));
                                    vbox1.getChildren().add(new Label("Grade: "
+(resultSet.getString("grade_4"))) );
                                    vbox1.getChildren().add(new Label("Subject Name: " +
(resultSet.getString("subject_5"))));
                                    vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_5"))));
                                    vbox1.getChildren().add(new Label("------------------------
________________________________"));
                                    // }
                                }else {

                                    System.out.println("No data found in the database.");
```

```
                        }
                    } catch (SQLException ex) {
                        ex.printStackTrace();
                    }


                    try(Connection con1 = DriverManager.getConnection(url,
user, password);
                            PreparedStatement preparedStatement =
con1.prepareStatement("INSERT INTO semester2(subject_1, grade_1, subject_2, grade_2,
subject_3, grade_3, subject_4, grade_4, subject_5, grade_5)VALUES (?, ?, ?, ?, ?, ?,
?,?,?,?)")){
                        Statement statement1 = con1.createStatement();

                        ResultSet resultSet = statement1.executeQuery("SELECT *FROM
semester2");
                        if (resultSet.next()) {

                            EventHandler <ActionEvent> consult1=new
EventHandler<ActionEvent>(){
                                public void handle(ActionEvent e){
                                    stage.setScene(sc112);
                                }
                            };
                            back.addEventHandler(ActionEvent.ACTION, consult1);
                            //while (resultSet.next()) { vbox1.getChildren().add(new
                            Label("SEMESTER-2"));
                            vbox1.getChildren().add(new Label("Subject NAME: " +
(resultSet.getString("subject_1"))));
                            vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_1"))));
                            vbox1.getChildren().add(new Label("Subject Name: "
+(resultSet.getString("subject_2")) ));
                            vbox1.getChildren().add(new Label("Grade: "
+(resultSet.getString("grade_2"))) );
                            vbox1.getChildren().add(new Label("Subject Name: " +
(resultSet.getString("subject_3"))));
                            vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_3"))));
                            vbox1.getChildren().add(new Label("Subject Name: "
+(resultSet.getString("subject_4")) ));
                            vbox1.getChildren().add(new Label("Grade: "
+(resultSet.getString("grade_4"))) );
```

```
                                    vbox1.getChildren().add(new Label("Subject Name: " +
(resultSet.getString("subject_5"))));
                                    vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_5"))));
                                    vbox1.getChildren().add(new Label("------------------------
______________________________"));
                                  // }
                                }else {

                                    System.out.println("No data found in the database.");
                                }
                            } catch (SQLException ex) {
                                ex.printStackTrace();
                            }
                            try(Connection con1 = DriverManager.getConnection(url,
user, password);
                                    PreparedStatement preparedStatement =
con1.prepareStatement("INSERT INTO semester3(subject_1, grade_1, subject_2, grade_2,
subject_3, grade_3, subject_4, grade_4, subject_5, grade_5)VALUES (?, ?, ?, ?, ?, ?,
?,?,?,?)")){
                                Statement statement1 = con1.createStatement();

                                ResultSet resultSet = statement1.executeQuery("SELECT *FROM
semester3");
                                if (resultSet.next()) {

                                    EventHandler <ActionEvent> consult1=new
EventHandler<ActionEvent>(){
                                        public void handle(ActionEvent e){
                                            stage.setScene(sc112);
                                        }
                                    };
                                    back.addEventHandler(ActionEvent.ACTION, consult1);
                                   // while (resultSet.next()) {
                                    vbox1.getChildren().add(new Label("SEMESTER -2" ));
                                    vbox1.getChildren().add(new Label("Subject NAME: " +
(resultSet.getString("subject_1"))));
                                    vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_1"))));
                                    vbox1.getChildren().add(new Label("Subject Name: "
+(resultSet.getString("subject_2")) ));
                                    vbox1.getChildren().add(new Label("Grade: "
+(resultSet.getString("grade_2"))) );
```

```
                                    vbox1.getChildren().add(new Label("Subject Name: " +
(resultSet.getString("subject_3"))));
                                    vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_3"))));
                                    vbox1.getChildren().add(new Label("Subject Name: "
+(resultSet.getString("subject_4")) ));
                                    vbox1.getChildren().add(new Label("Grade: "
+(resultSet.getString("grade_4"))) );
                                    vbox1.getChildren().add(new Label("Subject Name: " +
(resultSet.getString("subject_5"))));
                                    vbox1.getChildren().add(new Label("Grade: " +
(resultSet.getString("grade_5"))));
                                    vbox1.getChildren().add(new Label("-----------------------
______________________________"));
                                  // }
                                }else {

                                    System.out.println("No data found in the database.");
                                }
                            } catch (SQLException ex) {
                                ex.printStackTrace();
                            }

                            ScrollPane scrollPane1 = new ScrollPane();
                 scrollPane1.setContent(vbox1);
scrollPane1.setFitToWidth(true); // Adjusts the width of the scroll pane to thecontent
scrollPane1.setFitToHeight(true); // Adjusts the height of the scroll pane to thecontent

Scene sce1 = new Scene(scrollPane1, 500, 300);

               // Scene sce1=new Scene(vbox,500,300);
                stage.setScene(sce1);
                stage.show();
```

```
          }};
        but.addEventHandler(ActionEvent.ACTION, stud123);
       }};l2.addEventHandler(ActionEvent.ACTION, stud1);
       //}};
       //submit1.addEventHandler(ActionEvent.ACTION, reco);

       EventHandler<ActionEvent> stud = new EventHandler<ActionEvent>() {public void
         handle(ActionEvent e) {

        GridPane grid1=new GridPane(); Label
        l1=new Label("User Name");TextField
        tf=new TextField(); Label l2=new
        Label("Password"); TextField tf1=new
        TextField(); Button sub=new
        Button("Submit");Button back1=new
        Button("Back");grid1.add(l1,0,1);
        grid1.add(tf,1,1);
        grid1.add(l2,0,2);
        grid1.add(tf1,1,2);
        grid1.add(sub,0,3);
        grid1.add(back1,1,3);

        grid1.setStyle("-fx-background-color: orange;");
        EventHandler <ActionEvent> consult1=new
EventHandler<ActionEvent>(){
                              public void handle(ActionEvent e){
                                  stage.setScene(sc11);
                              }
                          };
                          back1.addEventHandler(ActionEvent.ACTION, consult1);Scene sc1=new
        Scene(grid1,350,200);
        stage.setTitle("Student Login page");
        stage.setScene(sc1);
        stage.show();
        EventHandler<ActionEvent> student1 = new
EventHandler<ActionEvent>() {
          public void handle(ActionEvent e) {VBox
        root1 = new VBox();
        Button btn1 = new Button("Student information");
        Button btn2 = new Button("Marks information");Button
        b1=new Button ("Back"); root1.setStyle("-fx-
        background-color: cyan;");
        root1.getChildren().add(btn1);
        root1.getChildren().add(btn2);
        root1.getChildren().add(b1); root1.setSpacing(10);
        root1.setAlignment(Pos.CENTER);
        EventHandler<ActionEvent> student1 = new
EventHandler<ActionEvent>() {
            public void handle(ActionEvent e) {
                stage.setScene(sc1);
            }
        };
        b1.addEventHandler(ActionEvent.ACTION, student1);Scene sce2 =
        new Scene(root1, 350, 200); stage.setTitle("Student Views");
        stage.setScene(sce2);
        stage.show();

        EventHandler<ActionEvent> student = new
EventHandler<ActionEvent>() {
            public void handle(ActionEvent e) {
                GridPane pane1 = new GridPane();
                Label rollno = new Label("Register Number");
                TextField fie = new TextField(); pane1.add(rollno, 0,
                0);
                pane1.add(fie, 1, 0);
                Label name = new Label("Name");
                TextField fie1 = new TextField();
                pane1.add(name, 0, 1);
                pane1.add(fie1, 1, 1);
                Label dob = new Label("Date of Birth");
                DatePicker da = new DatePicker();
                pane1.add(dob, 0, 2);
                pane1.add(da, 1, 2);
                Label age = new Label("Age");
                TextField fie2 = new TextField();
                pane1.add(age, 0, 3);
                pane1.add(fie2, 1, 3);
                ToggleGroup tgGender = new ToggleGroup();Label
                gender = new Label("Gender"); RadioButton r1 = new
                RadioButton("Male");
                r1.setToggleGroup(tgGender);
                RadioButton r2 = new RadioButton("Female");
                r2.setToggleGroup(tgGender);
                RadioButton r3 = new RadioButton("Other");
                r3.setToggleGroup(tgGender); pane1.add(gender, 0,
                4);
                pane1.add(r1, 1, 4);
                pane1.add(r2, 1, 5);
                pane1.add(r3, 1, 6);
                Label phone = new Label("Phone Number");
                TextField fie3 = new TextField(); pane1.add(phone,
                0, 7);
                pane1.add(fie3, 1, 7);
                Label city = new Label("City");
                TextField fie4 = new TextField();
                pane1.add(city, 0, 8);
                pane1.add(fie4, 1, 8);
                Label fees = new Label("Fees");
                TextField fie5 = new TextField();
                pane1.add(fees, 0, 9);
                pane1.add(fie5, 1, 9);
                Label dep = new Label("Department"); RadioButton
                rd1 = new RadioButton("CSE");
                rd1.setToggleGroup(tgGender);
                RadioButton rd2 = new RadioButton("ECE");
                rd2.setToggleGroup(tgGender);
                RadioButton rd3 = new RadioButton("EEE");
                rd3.setToggleGroup(tgGender); pane1.add(dep, 0,
                10);
                pane1.add(rd1, 1, 10);
                pane1.add(rd2, 1, 11);
                pane1.add(rd3, 1, 12);
                Button bt1 = new Button("Submit");
                Button bt2 = new Button("Back");
                pane1.add(bt1, 0, 13);
                pane1.add(bt2, 1, 13);
                Button v1 = new Button("View"); pane1.add(v1, 2,
                13); EventHandler<ActionEvent> back = new
        EventHandler<ActionEvent>() {
                    public void handle(ActionEvent e) {
                        stage.setScene(sce2);
                    }
                };

                bt2.addEventHandler(ActionEvent.ACTION, back);

                bt1.setOnAction(event -> {
                    String registerNumber = fie.getText();
                    String studentName = fie1.getText();
                    String dateOfBirth = (da.getValue() != null) ?
da.getValue().toString() : "";
                    String studentAge = fie2.getText(); String
                    studentGender = ((RadioButton)
tgGender.getSelectedToggle()).getText();
                    String phoneNumber = fie3.getText();String
                    studentCity = fie4.getText(); String
                    studentFees = fie5.getText();
                    String studentDepartment = ((RadioButton)
tgGender.getSelectedToggle()).getText();

                    try {
                        insertStudentData(registerNumber, studentName, dateOfBirth,
studentAge, studentGender, phoneNumber, studentCity, studentFees, studentDepartment);
                    } catch (SQLException ex) {
                        ex.printStackTrace();
                    }
                });
                EventHandler<ActionEvent> view3 = new
EventHandler<ActionEvent>() {
            public void handle(ActionEvent e) { String
                registerNumber = fie.getText();
                    String studentName = fie1.getText();
                    String dateOfBirth = (da.getValue() != null) ?
da.getValue().toString() : "";
                    String studentAge = fie2.getText(); String
                    studentGender = ((RadioButton)
tgGender.getSelectedToggle()).getText();
                    String phoneNumber = fie3.getText();String
                    studentCity = fie4.getText(); String
                    studentFees = fie5.getText();
                    String studentDepartment = ((RadioButton)
tgGender.getSelectedToggle()).getText();
                    VBox vbox1=new VBox();
                    vbox1.getChildren().add(new Label("Register
Number:"+(registerNumber)));
                     vbox1.getChildren().add(new
Label("StudentName:"+(studentName)));
                     vbox1.getChildren().add(new Label("Date of
Birth:"+(dateOfBirth)));
                      vbox1.getChildren().add(new Label("Age:"+(studentAge)));
                        vbox1.getChildren().add(new
Label("Gender:"+(studentGender)));
                      vbox1.getChildren().add(new Label("Phone
Number:"+(phoneNumber)));
                      vbox1.getChildren().add(new Label("City:"+(studentCity)));
                       vbox1.getChildren().add(new Label("Fees:"+(studentFees)));
                       vbox1.getChildren().add(new
Label("Dept:"+(studentDepartment)));
                    Alert dialog = new Alert(Alert.AlertType.INFORMATION);
                             dialog.setTitle("student mark view");
                             dialog.setHeaderText(null);
                             DialogPane dialogPane = dialog.getDialogPane();
                             dialogPane.setContent(vbox1); dialog.showAndWait();
            }
                };
            v1.addEventHandler(ActionEvent.ACTION, view3);Scene sce1 =
                new Scene(pane1, 450, 300); stage.setTitle("Student
                Information"); stage.setScene(sce1);
                stage.show();
            }
        };
        btn1.addEventHandler(ActionEvent.ACTION, student);

        EventHandler<ActionEvent> mark = new EventHandler<ActionEvent>() {public void
            handle(ActionEvent e) {
                VBox v1 = new VBox();
                Button s1 = new Button("Semester 1");
                Button s2 = new Button("Semester 2");
                Button s3 = new Button("Semester 3");
                Button back =new Button("Back");
                v1.getChildren().add(s1);
                v1.getChildren().add(s2);
                v1.getChildren().add(s3); v1.setSpacing(10);
                v1.setAlignment(Pos.CENTER); v1.getChildren().add(back);
                EventHandler<ActionEvent> back1 = new
EventHandler<ActionEvent>() {
                    public void handle(ActionEvent e) {
                        stage.setScene(sce2);
                    }
                };
                back.addEventHandler(ActionEvent.ACTION, back1);Scene
                sce4 = new Scene(v1, 350, 200); stage.setTitle("Marks
                Information"); stage.setScene(sce4);
                stage.show();

                EventHandler<ActionEvent> mark1 = new
EventHandler<ActionEvent>() {
                    public void handle(ActionEvent e) {
                        GridPane pane2 = new GridPane();
                        Label sub1 = new Label("Professional English - I");
                        ToggleGroup tgGrade1 = new ToggleGroup(); RadioButton
                        grade1Aplus = new RadioButton("A+");RadioButton grade1A
                        = new RadioButton("A"); RadioButton grade1Bplus = new
                        RadioButton("B+");RadioButton grade1B = new
                        RadioButton("B"); RadioButton grade1RA = new
                        RadioButton("RA"); grade1Aplus.setToggleGroup(tgGrade1);
                        grade1A.setToggleGroup(tgGrade1);
                        grade1Bplus.setToggleGroup(tgGrade1);
                        grade1B.setToggleGroup(tgGrade1);
                        grade1RA.setToggleGroup(tgGrade1);
                        pane2.add(grade1Aplus, 1, 0);
                        pane2.add(grade1A, 2, 0);
                        pane2.add(grade1Bplus, 3, 0);
                        pane2.add(grade1B, 4, 0);
                        pane2.add(grade1RA, 5, 0);
                        pane2.add(sub1, 0, 0);
                        Label sub2 = new Label("Matrices and Calculus");
                        ToggleGroup tgGrade2 = new ToggleGroup(); RadioButton
                        grade2Aplus = new RadioButton("A+");RadioButton grade2A
                        = new RadioButton("A"); RadioButton grade2Bplus = new
                        RadioButton("B+");RadioButton grade2B = new
                        RadioButton("B"); RadioButton grade2RA = new
                        RadioButton("RA");
```

```
grade2Aplus.setToggleGroup(tgGrade2);
grade2A.setToggleGroup(tgGrade2);
grade2Bplus.setToggleGroup(tgGrade2);
grade2B.setToggleGroup(tgGrade2);
grade2RA.setToggleGroup(tgGrade2);
pane2.add(grade2Aplus, 1, 1);
pane2.add(grade2A, 2, 1);
pane2.add(grade2Bplus, 3, 1);
pane2.add(grade2B, 4, 1);
pane2.add(grade2RA, 5, 1);
pane2.add(sub2, 0, 1);
Label sub3 = new Label("Engineering Physics");
pane2.add(sub3, 0, 2);
ToggleGroup tgGrade3 = new ToggleGroup(); RadioButton
grade3Aplus = new RadioButton("A+");RadioButton grade3A
= new RadioButton("A"); RadioButton grade3Bplus = new
RadioButton("B+");RadioButton grade3B = new
RadioButton("B"); RadioButton grade3RA = new
RadioButton("RA"); grade3Aplus.setToggleGroup(tgGrade3);
grade3A.setToggleGroup(tgGrade3);
grade3Bplus.setToggleGroup(tgGrade3);
grade3B.setToggleGroup(tgGrade3);
grade3RA.setToggleGroup(tgGrade3);
pane2.add(grade3Aplus, 1, 2);
pane2.add(grade3A, 2, 2);
pane2.add(grade3Bplus, 3, 2);
pane2.add(grade3B, 4, 2);
pane2.add(grade3RA, 5, 2);
Label sub4 = new Label("Engineering Chemistry");
pane2.add(sub4, 0, 3);
ToggleGroup tgGrade4 = new ToggleGroup(); RadioButton
grade4Aplus = new RadioButton("A+");RadioButton grade4A
= new RadioButton("A"); RadioButton grade4Bplus = new
RadioButton("B+");RadioButton grade4B = new
RadioButton("B"); RadioButton grade4RA = new
RadioButton("RA"); grade4Aplus.setToggleGroup(tgGrade4);
grade4A.setToggleGroup(tgGrade4);
grade4Bplus.setToggleGroup(tgGrade4);
grade4B.setToggleGroup(tgGrade4);
grade4RA.setToggleGroup(tgGrade4);
pane2.add(grade4Aplus, 1, 3);
```

```
                        pane2.add(grade4A, 2, 3);
                        pane2.add(grade4Bplus, 3, 3);
                        pane2.add(grade4B, 4, 3);
                        pane2.add(grade4RA, 5, 3);
                        Label sub5 = new Label("Problem Solving and Python
Programming");
                        pane2.add(sub5, 0, 4);
                        ToggleGroup tgGrade5 = new ToggleGroup(); RadioButton
                        grade5Aplus = new RadioButton("A+");RadioButton grade5A
                        = new RadioButton("A"); RadioButton grade5Bplus = new
                        RadioButton("B+");RadioButton grade5B = new
                        RadioButton("B"); RadioButton grade5RA = new
                        RadioButton("RA"); grade5Aplus.setToggleGroup(tgGrade5);
                        grade5A.setToggleGroup(tgGrade5);
                        grade5Bplus.setToggleGroup(tgGrade5);
                        grade5B.setToggleGroup(tgGrade5);
                        grade5RA.setToggleGroup(tgGrade5);
                        pane2.add(grade5Aplus, 1, 4);
                        pane2.add(grade5A, 2, 4);
                        pane2.add(grade5Bplus, 3, 4);
                        pane2.add(grade5B, 4, 4);
                        pane2.add(grade5RA, 5, 4);
                        Button submitGrades = new Button("Submit");Button
                        backButton = new Button("Back");
                        pane2.add(submitGrades, 0, 5);
                        pane2.add(backButton, 1, 5);
                        Button view1=new Button("Views");
                        pane2.add(view1,2,5);
                        pane2.setStyle("-fx-background-color: yellow;");
                        submitGrades.setOnAction(new EventHandler<ActionEvent>()
{
                            public void handle(ActionEvent e) {
                              //     String     roll=rollno.grtText();
                                String       subje1=sub1.getText();
                                String       subje2=sub2.getText();
                                String       subje3=sub3.getText();
                                String       subje4=sub4.getText();
                                String       subje5=sub5.getText();
                                String grad1 = ((RadioButton)
tgGrade1.getSelectedToggle()).getText();
                                String grad2 = ((RadioButton)
tgGrade2.getSelectedToggle()).getText();
```

```
                                String grad3 = ((RadioButton)
tgGrade3.getSelectedToggle()).getText();
                                String grad4 = ((RadioButton)
tgGrade4.getSelectedToggle()).getText();
                                String grad5 = ((RadioButton)
tgGrade5.getSelectedToggle()).getText();
                                 try {
                                insertMarksData( subje1, ((RadioButton)
tgGrade1.getSelectedToggle()).getText(), subje2, grad2, subje3,grad3,
subje4,  grad4,  subje5, grad5);
                          } catch (SQLException ex) {
                         ex.getMessage();
                    }
                                double gpa = calculateGPA(new String[]{grad1, grad2,
grad3, grad4, grad5});
                                System.out.println("Calculated GPA: " + gpa);
                                Alert alert = new Alert(Alert.AlertType.INFORMATION);
                                alert.setTitle("GPA Calculation"); alert.setHeaderText(null);
                                alert.setContentText("Calculated GPA: " + gpa);
                                alert.showAndWait();
                            }
                             });
                             EventHandler<ActionEvent> mar1 = new
EventHandler<ActionEvent>() {
                                public void handle(ActionEvent e) {String
                                grad1 = ((RadioButton)
tgGrade1.getSelectedToggle()).getText();
                                String grad2 = ((RadioButton)
tgGrade2.getSelectedToggle()).getText();
                                String grad3 = ((RadioButton)
tgGrade3.getSelectedToggle()).getText();
                                String grad4 = ((RadioButton)
tgGrade4.getSelectedToggle()).getText();
                                String grad5 = ((RadioButton)
tgGrade5.getSelectedToggle()).getText();
                                 VBox vbox = new VBox();
                                vbox.getChildren().add(new Label("Proffesional English -1:
" + (grad1)));
                                vbox.getChildren().add(new Label("Matrics and Calculus: "
+ (grad2)));
                               vbox.getChildren().add(new Label("Engineering Physics: "
+ (grad3)));
```

```
                                vbox.getChildren().add(new Label("Engineering Chemistry:
" + (grad4)));
                              vbox.getChildren().add(new Label("Problem Solving and
Python Programming: " + (grad5)));
                               Alert dialog = new Alert(Alert.AlertType.INFORMATION);
                             dialog.setTitle("student mark view"); dialog.setHeaderText(null);
                             DialogPane dialogPane = dialog.getDialogPane();
                             dialogPane.setContent(vbox); dialog.showAndWait();
                           }
                      };
                      view1.addEventHandler(ActionEvent.ACTION, mar1);
                       backButton.setOnAction(event -> {
                           stage.setScene(sce4);
                       });

                       Scene sce3 = new Scene(pane2, 500, 200);
                       stage.setTitle("Marks Information");
                       stage.setScene(sce3);
                       stage.show();
                   }
               };

               s1.addEventHandler(ActionEvent.ACTION, mark1);
               EventHandler<ActionEvent> mark2 = new
EventHandler<ActionEvent>() {
                                public void handle(ActionEvent e) {
                        GridPane pane3 = new GridPane();
                        //Label l1=new Label("Register Number");

                       Label sub1 = new Label("Professional English - 2");
                       ToggleGroup tgGrade11 = new ToggleGroup(); RadioButton
                       grade1Aplus = new RadioButton("A+");RadioButton grade1A
                       = new RadioButton("A"); RadioButton grade1Bplus = new
                       RadioButton("B+");RadioButton grade1B = new
                       RadioButton("B"); RadioButton grade1RA = new
                       RadioButton("RA");
                       grade1Aplus.setToggleGroup(tgGrade11);
                       grade1A.setToggleGroup(tgGrade11);
                       grade1Bplus.setToggleGroup(tgGrade11);
                       grade1B.setToggleGroup(tgGrade11);
                       grade1RA.setToggleGroup(tgGrade11);
```

```
pane3.add(grade1Aplus, 1, 0);
pane3.add(grade1A, 2, 0);
pane3.add(grade1Bplus, 3, 0);
pane3.add(grade1B, 4, 0);
pane3.add(grade1RA, 5, 0);
pane3.add(sub1, 0, 0);
Label sub2 = new Label("Statistics Methods"); ToggleGroup
tgGrade21 = new ToggleGroup(); RadioButton grade2Aplus =
new RadioButton("A+");RadioButton grade2A = new
RadioButton("A"); RadioButton grade2Bplus = new
RadioButton("B+");RadioButton grade2B = new
RadioButton("B"); RadioButton grade2RA = new
RadioButton("RA");
grade2Aplus.setToggleGroup(tgGrade21);
grade2A.setToggleGroup(tgGrade21);
grade2Bplus.setToggleGroup(tgGrade21);
grade2B.setToggleGroup(tgGrade21);
grade2RA.setToggleGroup(tgGrade21);
pane3.add(grade2Aplus, 1, 1);
pane3.add(grade2A, 2, 1);
pane3.add(grade2Bplus, 3, 1);
pane3.add(grade2B, 4, 1);
pane3.add(grade2RA, 5, 1);
pane3.add(sub2, 0, 1);
Label sub3 = new Label(" Information Science");
pane3.add(sub3, 0, 2);
ToggleGroup tgGrade31 = new ToggleGroup(); RadioButton
grade3Aplus = new RadioButton("A+");RadioButton grade3A
= new RadioButton("A"); RadioButton grade3Bplus = new
RadioButton("B+");RadioButton grade3B = new
RadioButton("B"); RadioButton grade3RA = new
RadioButton("RA");
grade3Aplus.setToggleGroup(tgGrade31);
grade3A.setToggleGroup(tgGrade31);
grade3Bplus.setToggleGroup(tgGrade31);
grade3B.setToggleGroup(tgGrade31);
grade3RA.setToggleGroup(tgGrade31);
pane3.add(grade3Aplus, 1, 2);
pane3.add(grade3A, 2, 2);
pane3.add(grade3Bplus, 3, 2);
pane3.add(grade3B, 4, 2);
pane3.add(grade3RA, 5, 2);
Label sub4 = new Label("Engineering Graphics");
```

```
pane3.add(sub4, 0, 3);
ToggleGroup tgGrade41 = new ToggleGroup(); RadioButton
grade4Aplus = new RadioButton("A+");RadioButton grade4A
= new RadioButton("A"); RadioButton grade4Bplus = new
RadioButton("B+");RadioButton grade4B = new
RadioButton("B"); RadioButton grade4RA = new
RadioButton("RA");
grade4Aplus.setToggleGroup(tgGrade41);
grade4A.setToggleGroup(tgGrade41);
grade4Bplus.setToggleGroup(tgGrade41);
grade4B.setToggleGroup(tgGrade41);
grade4RA.setToggleGroup(tgGrade41);
pane3.add(grade4Aplus, 1, 3);
pane3.add(grade4A, 2, 3);
pane3.add(grade4Bplus, 3, 3);
pane3.add(grade4B, 4, 3);
pane3.add(grade4RA, 5, 3);
Label sub5 = new Label("Programming in C");
pane3.add(sub5, 0, 4);
ToggleGroup tgGrade51 = new ToggleGroup(); RadioButton
grade5Aplus = new RadioButton("A+");RadioButton grade5A
= new RadioButton("A"); RadioButton grade5Bplus = new
RadioButton("B+");RadioButton grade5B = new
RadioButton("B"); RadioButton grade5RA = new
RadioButton("RA");
grade5Aplus.setToggleGroup(tgGrade51);
grade5A.setToggleGroup(tgGrade51);
grade5Bplus.setToggleGroup(tgGrade51);
grade5B.setToggleGroup(tgGrade51);
grade5RA.setToggleGroup(tgGrade51);
pane3.add(grade5Aplus, 1, 4);
pane3.add(grade5A, 2, 4);
pane3.add(grade5Bplus, 3, 4);
pane3.add(grade5B, 4, 4);
pane3.add(grade5RA, 5, 4);
Label l1=new Label("Register Name");
TextField text =new TextField();
pane3.add(l1,0,5);
pane3.add(text,1,5);
Button submitGrades1 = new Button("Submit");Button
backButton = new Button("Back");
pane3.add(submitGrades1, 0, 6);
pane3.add(backButton, 1, 6);
```

```
                        Button view1=new Button("Views");
                        pane3.add(view1,2,6);
                        pane3.setStyle("-fx-background-color: pink;");
                        submitGrades1.setOnAction(new EventHandler<ActionEvent>()
{
                            public void handle(ActionEvent e) {String
                                c1=sub1.getText();
                                String c2=sub2.getText(); String
                                c3=sub3.getText(); String
                                c4=sub4.getText(); String
                                c5=sub5.getText(); String mark1 =
                                ((RadioButton)
tgGrade11.getSelectedToggle()).getText();
                                String mark2 = ((RadioButton)
tgGrade21.getSelectedToggle()).getText();
                                String mark3 = ((RadioButton)
tgGrade31.getSelectedToggle()).getText();
                                String mark4 = ((RadioButton)
tgGrade41.getSelectedToggle()).getText();
                                String mark5 = ((RadioButton)
tgGrade51.getSelectedToggle()).getText();

                                 try {
                                insertMarksData1( c1, mark1, c2, mark2, c3,mark3,
c4,  mark4, c5, mark5);
                        } catch (SQLException ex) {
                        ex.printStackTrace();
                    }

                                double gpa = calculateGPA(new String[]{mark1, mark2,mark3,
mark4, mark5});

                                System.out.println("Calculated GPA: " + gpa);

                                Alert alert = new Alert(Alert.AlertType.INFORMATION);
                                alert.setTitle("GPA Calculation"); alert.setHeaderText(null);
                                alert.setContentText("Calculated GPA: " + gpa);


                                alert.showAndWait();
                            }
                        });
                         backButton.setOnAction(event -> {
```

```
                    stage.setScene(sce4);
                  });
                  EventHandler<ActionEvent> mark2 = new
EventHandler<ActionEvent>() {
                       public void handle(ActionEvent e) {

                       String grad1 = ((RadioButton)
tgGrade11.getSelectedToggle()).getText();
                       String grad2 = ((RadioButton)
tgGrade21.getSelectedToggle()).getText();
                       String grad3 = ((RadioButton)
tgGrade31.getSelectedToggle()).getText();
                       String grad4 = ((RadioButton)
tgGrade41.getSelectedToggle()).getText();
                       String grad5 = ((RadioButton)
tgGrade51.getSelectedToggle()).getText();
                        VBox vbox = new VBox();
                       vbox.getChildren().add(new Label("Proffesional English -2:
" + (grad1)));
                       vbox.getChildren().add(new Label("Statistics Methods: " +
(grad2)));
                      vbox.getChildren().add(new Label("Information Science: " +
(grad3)));
                      vbox.getChildren().add(new Label("Engineering Graphics:
" + (grad4)));
                      vbox.getChildren().add(new Label("Programming in C :" +
(grad5)));
                       Alert dialog = new Alert(Alert.AlertType.INFORMATION);
                     dialog.setTitle("student mark view"); dialog.setHeaderText(null);
                     DialogPane dialogPane = dialog.getDialogPane();
                     dialogPane.setContent(vbox); dialog.showAndWait();
                    }
                 };
                 view1.addEventHandler(ActionEvent.ACTION, mark2);



                  Scene sce3 = new Scene(pane3, 500, 200);
                  stage.setTitle("Marks Information");
                  stage.setScene(sce3);
                  stage.show();
```

```
                    }
                };
                s2.addEventHandler(ActionEvent.ACTION, mark2);
                EventHandler<ActionEvent> sem3 = new
EventHandler<ActionEvent>() {
                        public void handle(ActionEvent e) {
                     GridPane pane4 = new GridPane();
                    Label sub1 = new Label("Discrete Mathematics");
                    ToggleGroup tgGrade113 = new ToggleGroup(); RadioButton
                    grade1Aplus = new RadioButton("A+");RadioButton grade1A
                    = new RadioButton("A"); RadioButton grade1Bplus = new
                    RadioButton("B+");RadioButton grade1B = new
                    RadioButton("B"); RadioButton grade1RA = new
                    RadioButton("RA");
                    grade1Aplus.setToggleGroup(tgGrade113);
                    grade1A.setToggleGroup(tgGrade113);
                    grade1Bplus.setToggleGroup(tgGrade113);
                    grade1B.setToggleGroup(tgGrade113);
                    grade1RA.setToggleGroup(tgGrade113);
                    pane4.add(grade1Aplus, 1, 0);
                    pane4.add(grade1A, 2, 0);
                    pane4.add(grade1Bplus, 3, 0);
                    pane4.add(grade1B, 4, 0);
                    pane4.add(grade1RA, 5, 0);
                    pane4.add(sub1, 0, 0);
                    Label sub2 = new Label("Foundations of Data Science");
                    ToggleGroup tgGrade213 = new ToggleGroup(); RadioButton
                    grade2Aplus = new RadioButton("A+"); RadioButton grade2A =
                    new RadioButton("A"); RadioButton grade2Bplus = new
                    RadioButton("B+"); RadioButton grade2B = new
                    RadioButton("B"); RadioButton grade2RA = new
                    RadioButton("RA"); grade2Aplus.setToggleGroup(tgGrade213);
                    grade2A.setToggleGroup(tgGrade213);
                    grade2Bplus.setToggleGroup(tgGrade213);
                    grade2B.setToggleGroup(tgGrade213);
                    grade2RA.setToggleGroup(tgGrade213); pane4.add(grade2Aplus,
                    1, 1);
                    pane4.add(grade2A, 2, 1);
                    pane4.add(grade2Bplus, 3, 1);
                    pane4.add(grade2B, 4, 1);
                    pane4.add(grade2RA, 5, 1);
                    pane4.add(sub2, 0, 1);
```

```
Label sub3 = new Label("Data Structures");
pane4.add(sub3, 0, 2);
ToggleGroup tgGrade313 = new ToggleGroup(); RadioButton
grade3Aplus = new RadioButton("A+");RadioButton grade3A
= new RadioButton("A"); RadioButton grade3Bplus = new
RadioButton("B+");RadioButton grade3B = new
RadioButton("B"); RadioButton grade3RA = new
RadioButton("RA");
grade3Aplus.setToggleGroup(tgGrade313);
grade3A.setToggleGroup(tgGrade313);
grade3Bplus.setToggleGroup(tgGrade313);
grade3B.setToggleGroup(tgGrade313);
grade3RA.setToggleGroup(tgGrade313);
pane4.add(grade3Aplus, 1, 2);
pane4.add(grade3A, 2, 2);
pane4.add(grade3Bplus, 3, 2);
pane4.add(grade3B, 4, 2);
pane4.add(grade3RA, 5, 2);
Label sub4 = new Label("Object Oriented Programming");
pane4.add(sub4, 0, 3);
ToggleGroup tgGrade413 = new ToggleGroup(); RadioButton
grade4Aplus = new RadioButton("A+");RadioButton grade4A
= new RadioButton("A"); RadioButton grade4Bplus = new
RadioButton("B+");RadioButton grade4B = new
RadioButton("B"); RadioButton grade4RA = new
RadioButton("RA");
grade4Aplus.setToggleGroup(tgGrade413);
grade4A.setToggleGroup(tgGrade413);
grade4Bplus.setToggleGroup(tgGrade413);
grade4B.setToggleGroup(tgGrade413);
grade4RA.setToggleGroup(tgGrade413);
pane4.add(grade4Aplus, 1, 3);
pane4.add(grade4A, 2, 3);
pane4.add(grade4Bplus, 3, 3);
pane4.add(grade4B, 4, 3);
pane4.add(grade4RA, 5, 3);
Label sub5 = new Label("Principles and Computer
Organization");
pane4.add(sub5, 0, 4);
ToggleGroup tgGrade513 = new ToggleGroup(); RadioButton
grade5Aplus = new RadioButton("A+");RadioButton grade5A
= new RadioButton("A"); RadioButton grade5Bplus = new
RadioButton("B+");
```

```
                        RadioButton grade5B = new RadioButton("B");
                        RadioButton grade5RA = new RadioButton("RA");
                        grade5Aplus.setToggleGroup(tgGrade513);
                        grade5A.setToggleGroup(tgGrade513);
                        grade5Bplus.setToggleGroup(tgGrade513);
                        grade5B.setToggleGroup(tgGrade513);
                        grade5RA.setToggleGroup(tgGrade513);
                        pane4.add(grade5Aplus, 1, 4);
                        pane4.add(grade5A, 2, 4);
                        pane4.add(grade5Bplus, 3, 4);
                        pane4.add(grade5B, 4, 4);
                        pane4.add(grade5RA, 5, 4);
                        Button submitGrades11 = new Button("Submit");Button
                        backButton = new Button("Back");
                        pane4.add(submitGrades11, 0, 5);
                        pane4.add(backButton, 1, 5);
                        Button view1=new Button("Views");
                        pane4.add(view1,2,5);
                        pane4.setStyle("-fx-background-color:magenta;");
                        submitGrades11.setOnAction(new
EventHandler<ActionEvent>() {
                            public void handle(ActionEvent e) {
                            //    String mark11=sub1.getText();
                                String c13=sub1.getText(); String
                                c23=sub2.getText(); String
                                c33=sub3.getText(); String
                                c43=sub4.getText(); String
                                c53=sub5.getText(); String mark13 =
                                ((RadioButton)
tgGrade113.getSelectedToggle()).getText();
                                String mark23 = ((RadioButton)
tgGrade213.getSelectedToggle()).getText();
                                String mark33 = ((RadioButton)
tgGrade313.getSelectedToggle()).getText();
                                String mark43 = ((RadioButton)
tgGrade413.getSelectedToggle()).getText();
                                String mark53 = ((RadioButton)
tgGrade513.getSelectedToggle()).getText();
                                 try {
                                insertMarksData2( c13, mark13, c23, mark23, c33,mark33,
c43, mark43, c53, mark53);
                        } catch (SQLException ex) {
                        ex.printStackTrace();
```

```
                    }

                                    double gpa = calculateGPA(new String[]{mark13, mark23,mark33,
mark43, mark53});

                                    System.out.println("Calculated GPA: " + gpa);

                                    Alert alert = new Alert(Alert.AlertType.INFORMATION);
                                    alert.setTitle("GPA Calculation"); alert.setHeaderText(null);
                                    alert.setContentText("Calculated GPA: " + gpa);

                                    alert.showAndWait();
                              }
                          });
                          backButton.setOnAction(event -> {
                              stage.setScene(sce4);
                          });
                          EventHandler<ActionEvent> mark23 = new
EventHandler<ActionEvent>() {
                                    public void handle(ActionEvent e) {

                                     String mark13 = ((RadioButton)
tgGrade113.getSelectedToggle()).getText();
                                    String mark23 = ((RadioButton)
tgGrade213.getSelectedToggle()).getText();
                                    String mark33 = ((RadioButton)
tgGrade313.getSelectedToggle()).getText();
                                    String mark43 = ((RadioButton)
tgGrade413.getSelectedToggle()).getText();
                                    String mark53 = ((RadioButton)
tgGrade513.getSelectedToggle()).getText();
                                     VBox vbox = new VBox();
                                    vbox.getChildren().add(new Label("Discrete Mathematics:
" + (mark13)));
                                    vbox.getChildren().add(new Label("Foundations of Data
Science: " + (mark23)));
                                   vbox.getChildren().add(new Label("Data Structures: " +
(mark33)));
                                    vbox.getChildren().add(new Label("Object Oriented
Programming:" + (mark43)));
                                  vbox.getChildren().add(new Label("Principles and Computer
Organization" + (mark53)));
```

```
                             Alert dialog = new Alert(Alert.AlertType.INFORMATION);
                           dialog.setTitle("student mark view"); dialog.setHeaderText(null);
                           DialogPane dialogPane = dialog.getDialogPane();
                           dialogPane.setContent(vbox); dialog.showAndWait();
                         }
                     };
                     view1.addEventHandler(ActionEvent.ACTION, mark23);



                       Scene sce3 = new Scene(pane4, 500, 200);
                       stage.setTitle("Marks Information");
                       stage.setScene(sce3);
                       stage.show();
                   }
               };
               s3.addEventHandler(ActionEvent.ACTION, sem3);
           }
       };
       btn2.addEventHandler(ActionEvent.ACTION, mark);

}};
  sub.addEventHandler(ActionEvent.ACTION, student1);
  }
  };
  l1.addEventHandler(ActionEvent.ACTION, stud);
  }
   private double calculateGPA(String[] grades) {int
       totalPoints = 0;
       for (String grade : grades) {
           switch (grade) {
               case "A+": totalPoints
                   += 10;break;
               case "A": totalPoints
                   += 9;break;
               case "B+": totalPoints
                   += 8;break;
               case "B":
                   totalPoints += 7;
                   break;
               case "RA": totalPoints
                   += 0;break;
           }
       }
       return totalPoints / (double) grades.length;
   }


   private void insertStudentData(String registerNumber, String name, Stringdob, String
age, String gender, String phone, String city, String fees, String department) throws
SQLException {
       try (Connection connection = DriverManager.getConnection(url, user,password))
{
           String query = "INSERT INTO studentinformation (Register_Number, Name,
Date_Of_Birth, Age, Gender, Phone_Number, City, Fees, Department) VALUES (?, ?, ?, ?,
?, ?, ?, ?, ?)";
           try (PreparedStatement statement =
connection.prepareStatement(query)) {
               statement.setString(1, registerNumber);
               statement.setString(2, name);
               statement.setString(3, dob);
               statement.setString(4, age);
               statement.setString(5, gender);
               statement.setString(6, phone);
               statement.setString(7, city);
               statement.setString(8, fees);
               statement.setString(9, department);
               statement.executeUpdate();
           }
       }
   }
   private void insertMarksData(String subject1, String grade1,String subject2,String
grade2, String subject3,String grade3, String subject4, String grade4, String subject5,String
grade5) throws SQLException {
       System.out.println("inside the details");
       try (Connection con = DriverManager.getConnection(url, user, password))
{
       String query = "INSERT INTO semester1(subject_1, Grade_1, subject_2,
grade_2, subject_3, grade_3, subject_4, grade_4, subject_5, grade_5) VALUES(?, ?, ?, ?, ?, ?,
?,?,?,?)";
```

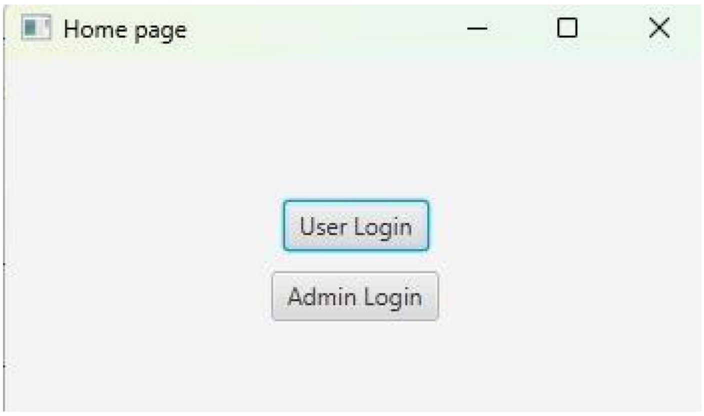
```
          try (PreparedStatement preparedStatement =
con.prepareStatement(query)){
        preparedStatement.setString(1, subject1);
        preparedStatement.setString(2, grade1);
        preparedStatement.setString(3, subject2);
        preparedStatement.setString(4, grade2);
        preparedStatement.setString(5, subject3);
        preparedStatement.setString(6, grade3);
        preparedStatement.setString(7, subject4);
        preparedStatement.setString(8, grade4);
        preparedStatement.setString(9, subject5);
        preparedStatement.setString(10, grade5);
        preparedStatement.executeUpdate();
          }
    }
    }
    private void insertMarksData1(String subject1, String grade1,String subject2,String
grade2, String subject3,String grade3, String subject4, String grade4, String subject5,String
grade5) throws SQLException {
        System.out.println("inside the details");
        try (Connection con = DriverManager.getConnection(url, user, password))
{
        String query = "INSERT INTO semester2(subject_1, grade_1, subject_2,
grade_2, subject_3, grade_3, subject_4, grade_4, subject_5, grade_5) VALUES(?, ?, ?, ?, ?, ?,
?,?,?,?)";
          try (PreparedStatement preparedStatement =
con.prepareStatement(query)){
        preparedStatement.setString(1, subject1);
        preparedStatement.setString(2, grade1);
        preparedStatement.setString(3, subject2);
        preparedStatement.setString(4, grade2);
        preparedStatement.setString(5, subject3);
        preparedStatement.setString(6, grade3);
        preparedStatement.setString(7, subject4);
        preparedStatement.setString(8, grade4);
        preparedStatement.setString(9, subject5);
        preparedStatement.setString(10, grade5);


         preparedStatement.executeUpdate();
          }
    }
    }
    private void insertMarksData2(String subject1, String grade1,String subject2,String
grade2, String subject3,String grade3, String subject4, String grade4, String subject5,String
grade5) throws SQLException {
        System.out.println("inside the details");
        try (Connection con = DriverManager.getConnection(url, user, password))
{
        String query = "INSERT INTO semester3(subject_1, grade_1, subject_2,
grade_2, subject_3, grade_3, subject_4, grade_4, subject_5, grade_5) VALUES(?, ?, ?, ?, ?, ?,
?,?,?,?)";
         try (PreparedStatement preparedStatement =
con.prepareStatement(query)){
        preparedStatement.setString(1, subject1);
        preparedStatement.setString(2, grade1);
        preparedStatement.setString(3, subject2);
        preparedStatement.setString(4, grade2);
        preparedStatement.setString(5, subject3);
        preparedStatement.setString(6, grade3);
        preparedStatement.setString(7, subject4);
        preparedStatement.setString(8, grade4);
        preparedStatement.setString(9, subject5);
        preparedStatement.setString(10, grade5);
        preparedStatement.executeUpdate();
          }
    }
    }
    public static void main(String[] args) {
        launch();
    }
}
```

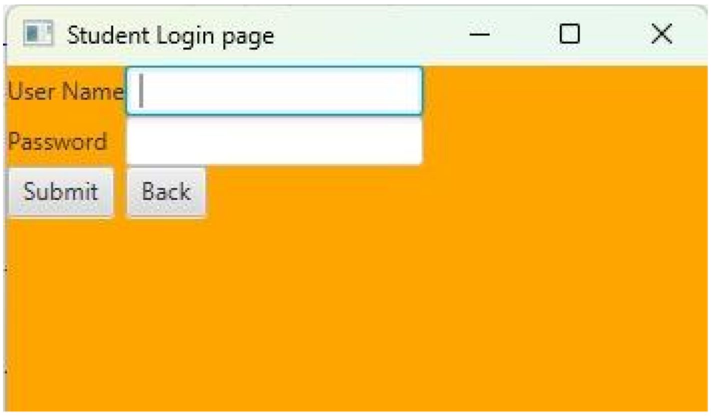
CHAPTER 5

SCREENSHOTS

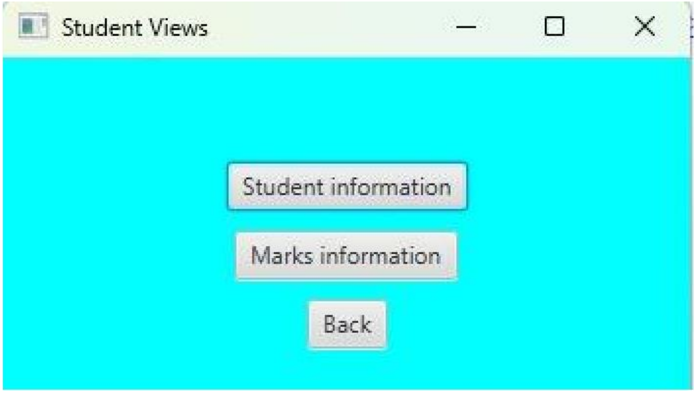
Initial Page:



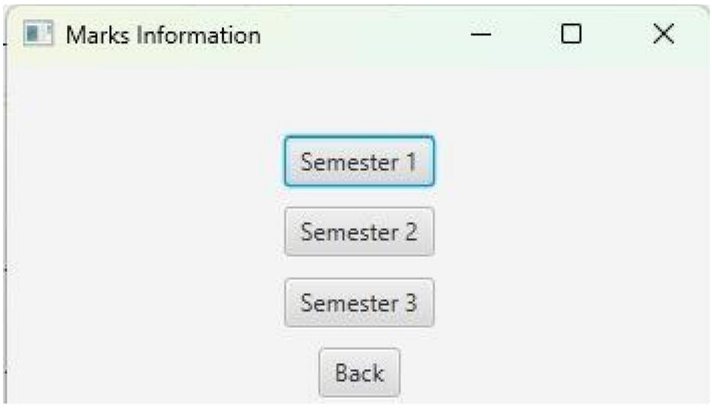
Student login page:



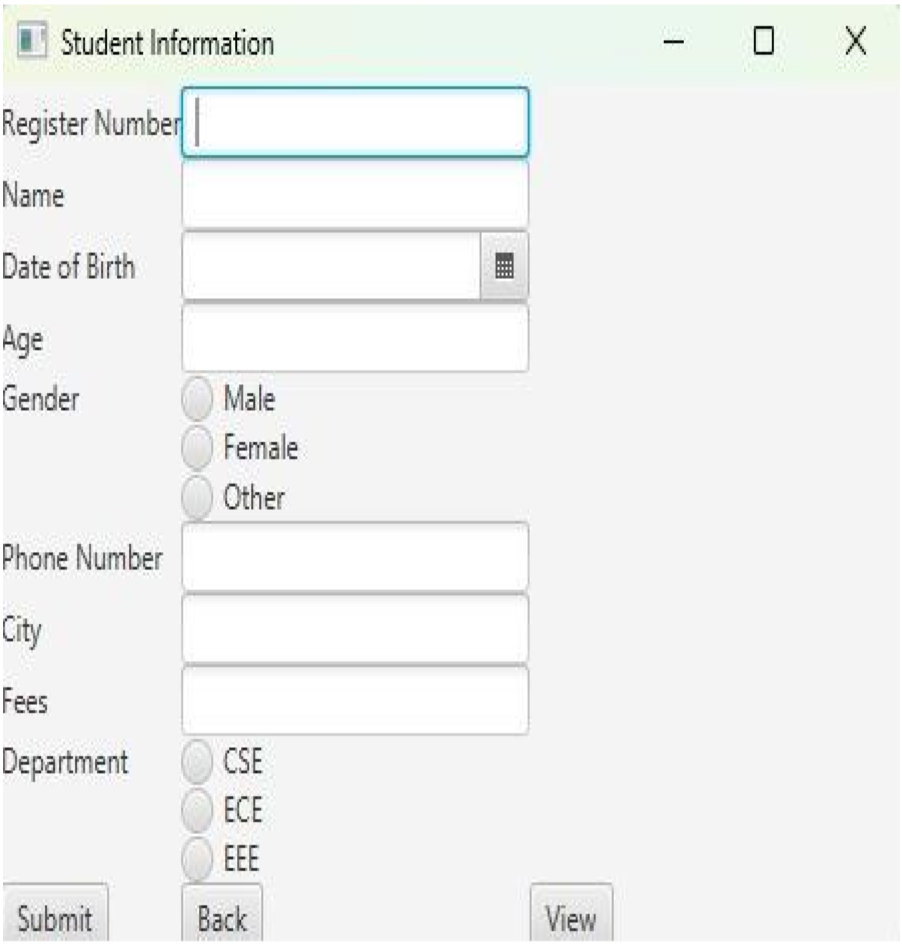
Student View:



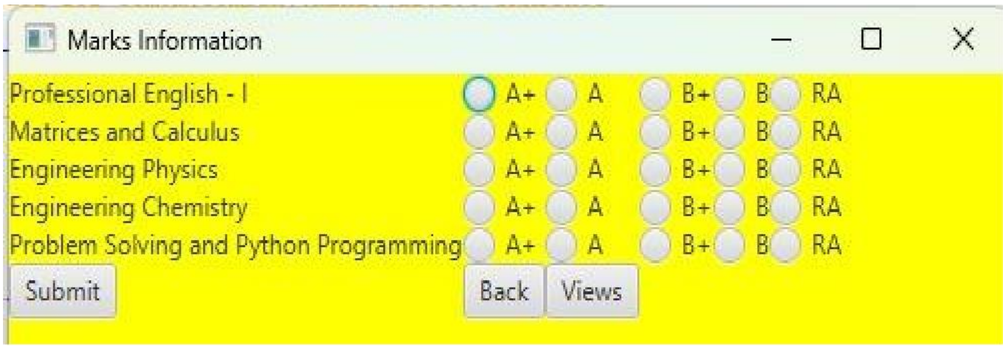
Marks page:



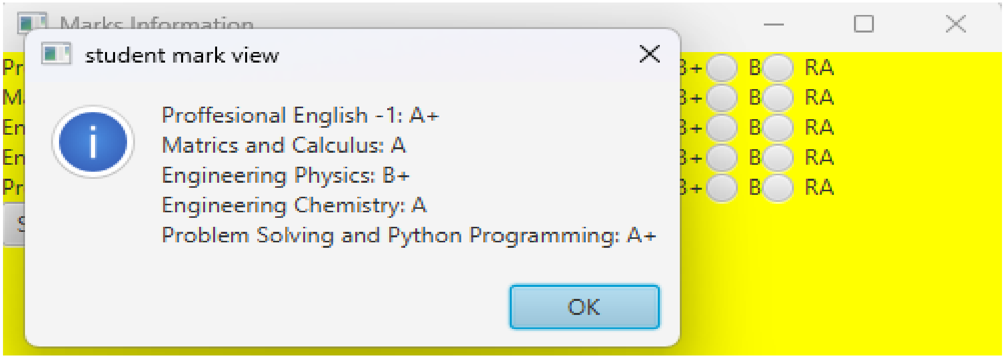
Student Information Page:



Semester I grade page:

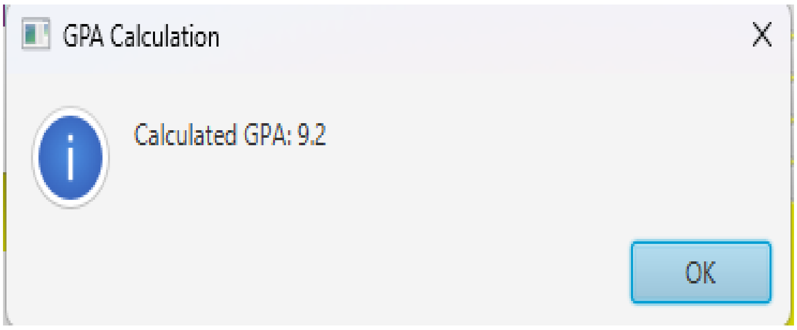

Marks Information
Professional English - I ○ A+ ○ A ○ B+ ○ B ○ RA
Matrices and Calculus ○ A+ ○ A ○ B+ ○ B ○ RA
Engineering Physics ○ A+ ○ A ○ B+ ○ B ○ RA
Engineering Chemistry ○ A+ ○ A ○ B+ ○ B ○ RA
Problem Solving and Python Programming ○ A+ ○ A ○ B+ ○ B ○ RA
Submit
Back
Views

Student mark view:



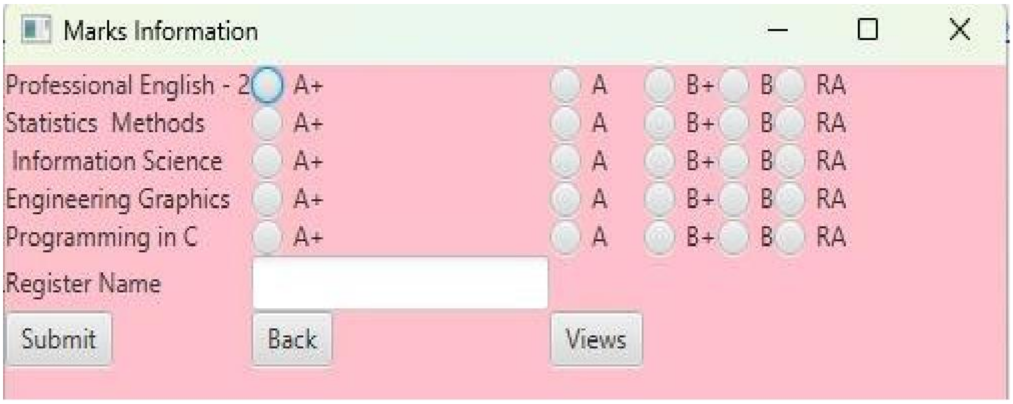
Proffesional English -1: A+
Matrics and Calculus: A
Engineering Physics: B+
Engineering Chemistry: A
Problem Solving and Python Programming: A+

GPA calculation:


GPA Calculation

Calculated GPA: 9.2

Semester II grade page:


Marks Information

Professional English - 2 ○ A+ ○ A ○ B+ ○ B ○ RA

Statistics Methods ○ A+ ○ A ○ B+ ○ B ○ RA

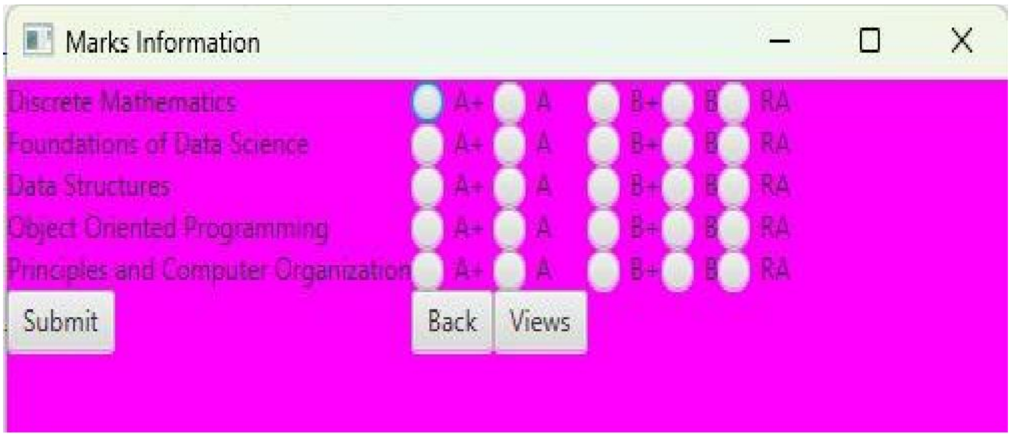
Information Science ○ A+ ○ A ○ B+ ○ B ○ RA

Engineering Graphics ○ A+ ○ A ○ B+ ○ B ○ RA

Programming in C ○ A+ ○ A ○ B+ ○ B ○ RA

Register Name

Semester III grade page:


Marks Information

Discrete Mathematics ○ A+ ○ A ○ B+ ○ B ○ RA

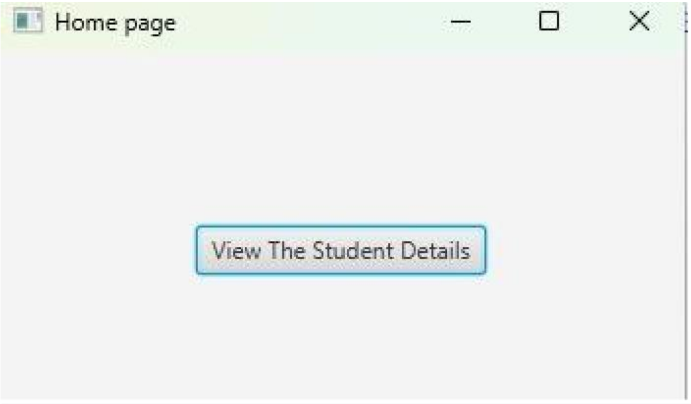
Foundations of Data Science ○ A+ ○ A ○ B+ ○ B ○ RA

Data Structures ○ A+ ○ A ○ B+ ○ B ○ RA

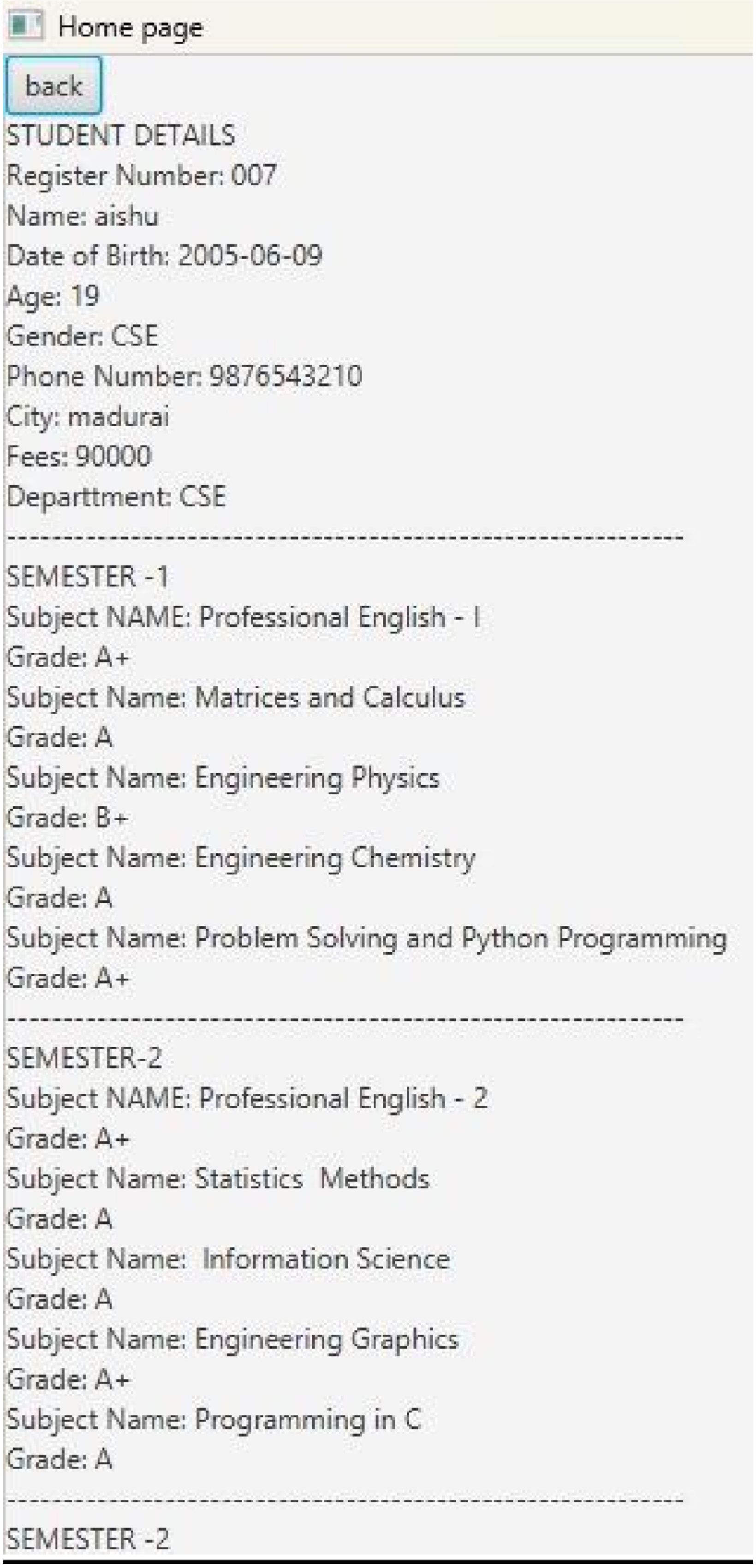
Object Oriented Programming ○ A+ ○ A ○ B+ ○ B ○ RA

Principles and Computer Organization ○ A+ ○ A ○ B+ ○ B ○ RA

Student view

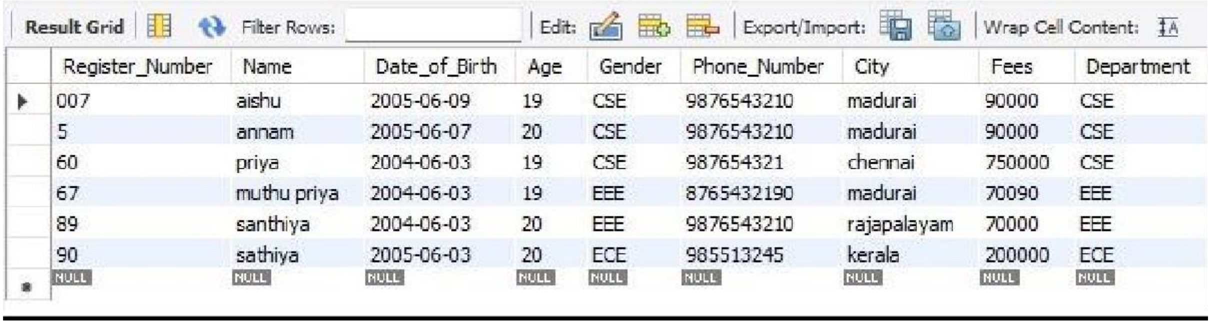

Home page

Student view:

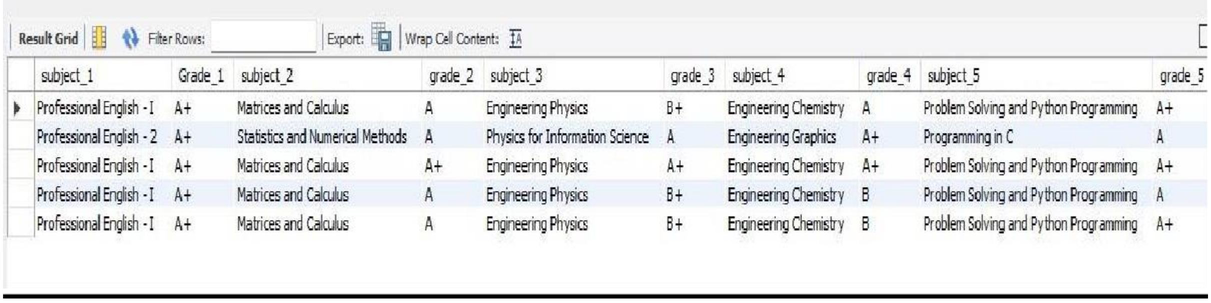


```
STUDENT DETAILS
Register Number: 007
Name: aishu
Date of Birth: 2005-06-09
Age: 19
Gender: CSE
Phone Number: 9876543210
City: madurai
Fees: 90000
Departtment: CSE
------------------------------------------------------------
SEMESTER -1
Subject NAME: Professional English - I
Grade: A+
Subject Name: Matrices and Calculus
Grade: A
Subject Name: Engineering Physics
Grade: B+
Subject Name: Engineering Chemistry
Grade: A
Subject Name: Problem Solving and Python Programming
Grade: A+
------------------------------------------------------------
SEMESTER-2
Subject NAME: Professional English - 2
Grade: A+
Subject Name: Statistics  Methods
Grade: A
Subject Name:  Information Science
Grade: A
Subject Name: Engineering Graphics
Grade: A+
Subject Name: Programming in C
Grade: A
------------------------------------------------------------
SEMESTER -2
```

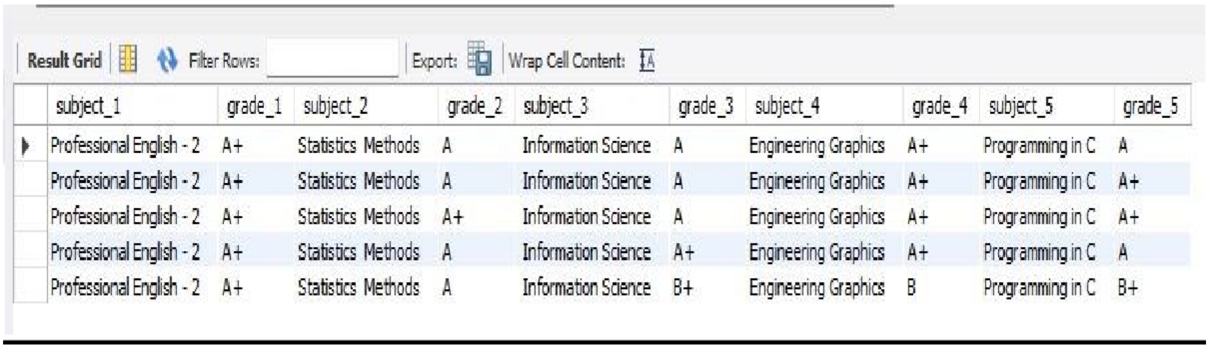
Stored information in DB:



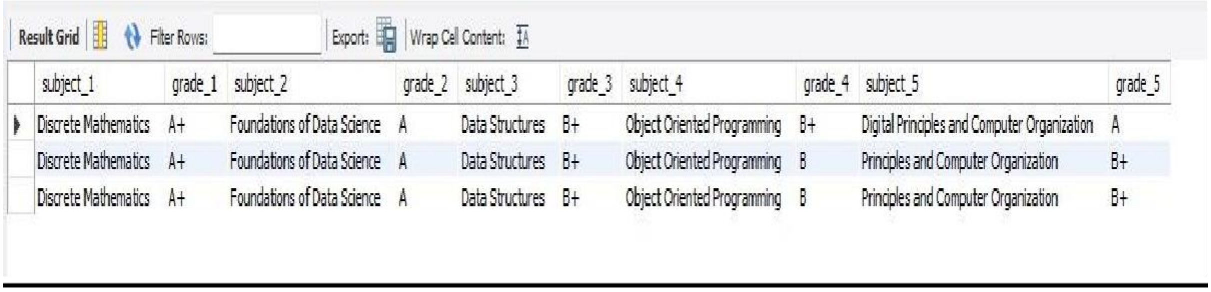
Register_Number	Name	Date_of_Birth	Age	Gender	Phone_Number	City	Fees	Department
007	aishu	2005-06-09	19	CSE	9876543210	madurai	90000	CSE
5	annam	2005-06-07	20	CSE	9876543210	madurai	90000	CSE
60	priya	2004-06-03	19	CSE	987654321	chennai	750000	CSE
67	muthu priya	2004-06-03	19	EEE	8765432190	madurai	70090	EEE
89	santhiya	2004-06-03	20	EEE	9876543210	rajapalayam	70000	EEE
90	sathiya	2005-06-03	20	ECE	985513245	kerala	200000	ECE
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



subject_1	Grade_1	subject_2	grade_2	subject_3	grade_3	subject_4	grade_4	subject_5	grade_5
Professional English - I	A+	Matrices and Calculus	A	Engineering Physics	B+	Engineering Chemistry	A	Problem Solving and Python Programming	A+
Professional English - 2	A+	Statistics and Numerical Methods	A	Physics for Information Science	A	Engineering Graphics	A+	Programming in C	A
Professional English - I	A+	Matrices and Calculus	A+	Engineering Physics	A+	Engineering Chemistry	A+	Problem Solving and Python Programming	A+
Professional English - I	A+	Matrices and Calculus	A	Engineering Physics	B+	Engineering Chemistry	B	Problem Solving and Python Programming	A
Professional English - I	A+	Matrices and Calculus	A	Engineering Physics	B+	Engineering Chemistry	B	Problem Solving and Python Programming	A+



subject_1	grade_1	subject_2	grade_2	subject_3	grade_3	subject_4	grade_4	subject_5	grade_5
Professional English - 2	A+	Statistics Methods	A	Information Science	A	Engineering Graphics	A+	Programming in C	A
Professional English - 2	A+	Statistics Methods	A	Information Science	A	Engineering Graphics	A+	Programming in C	A+
Professional English - 2	A+	Statistics Methods	A+	Information Science	A	Engineering Graphics	A+	Programming in C	A+
Professional English - 2	A+	Statistics Methods	A	Information Science	A+	Engineering Graphics	A+	Programming in C	A
Professional English - 2	A+	Statistics Methods	A	Information Science	B+	Engineering Graphics	B	Programming in C	B+



subject_1	grade_1	subject_2	grade_2	subject_3	grade_3	subject_4	grade_4	subject_5	grade_5
Discrete Mathematics	A+	Foundations of Data Science	A	Data Structures	B+	Object Oriented Programming	B+	Digital Principles and Computer Organization	A
Discrete Mathematics	A+	Foundations of Data Science	A	Data Structures	B+	Object Oriented Programming	B	Principles and Computer Organization	B+
Discrete Mathematics	A+	Foundations of Data Science	A	Data Structures	B+	Object Oriented Programming	B	Principles and Computer Organization	B+

CHAPTER 6
CONCLUSION:

The Student Record Management System (SRMS) represents a significant advancement in the management of student records within educational institutions. By centralizing student information, enhancing data integrity, and improving accessibility for users, the SRMS addresses critical challenges associated with traditional paper-based systems. The project has successfully demonstrated improvements in operational efficiency, data accuracy, and user satisfaction across different user roles including students, faculty, and administrators.

Challenges encountered during the development process have been effectively managed, paving the way for potential future enhancements such as mobile application integration and advanced analytics capabilities. Ultimately, the SRMS stands to enhance administrative processes, support informed decision- making, and contribute positively to the overall educational experience within in stitutions.

CHAPTER 7
FUTURE WORK

In the future, student records management will undergo a profound transformation driven by advanced technologies and evolving educational needs. Systems will integrate AI-powered analytics to provide predictive insights into student performance, allowing educators to personalize learning experiences effectively. Blockchain technology will secure and streamline the verification of academic credentials, ensuring their authenticity and accessibility worldwide. Virtual and augmented reality will revolutionize how students interact with their academic records, creating immersive learning environments where historical achievements and future goals are seamlessly integrated. Moreover, cloud-based platforms will enable real-time updates and collaboration among students, educators, and administrators, fostering a more connected and efficient educational ecosystem. As these innovations continue to evolve, student records management will not only become more efficient but also more responsive to the dynamic needs of students and educational institutions alike.

CHAPTER 8
REFERENCE

1.https://www.iitms.co.in/blog/student-record-management-system.html

2.https://www.academia.edu/28742043/Students_Record_Management_System_Project_By_Soita_Reuben

-

+

PERFORMANCE	
VIVAVOCE	
MINI PROJECT	
TOTAL	