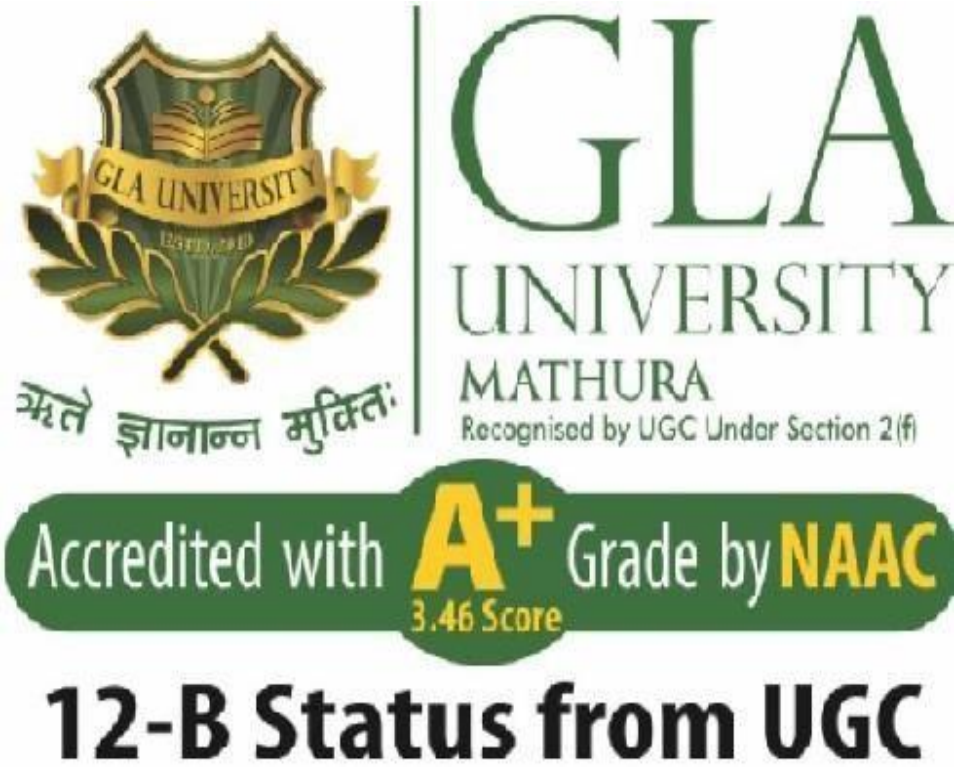
Student Registration Form

Master of Computer Applications

Department of Computer Engineering & Applications



GLA University
Mathura- 281406, INDIA

Name – Mani Mishra

Roll no – 2284200114(43)

Subject – C# .Net

CONTENTS

OBJECTIVE

The objective of this project is to develop a student registration system using C# programming language. The system aims to create an intuitive and user-friendly interface where users can input and manage student information efficiently. The primary goals include

Data Collection: Gather essential details such as student names, ages, genders, contact information, and academic specifics.

User Interaction: Provide a user interface that allows seamless input and modification of student data.

Data Validation: Implement validation checks to ensure the accuracy and integrity of the information entered by users.

BASIC INTRODUCTION

The Student Registration System is a software application developed in C#. This system is designed to simplify the process of registering students in educational institutions. It provides an easy-to-use interface for administrators to manage student information and for students to register for courses.

The main features of the system include:

1. **Student Information Management**: Administrators can add, update, and delete student information. This includes personal details, contact information, and academic records.
2. **Course Registration**: Students can register for courses offered by the institution. The system checks for course prerequisites and availability before confirming the registration.
3. **Report Generation**: The system can generate various reports such as a list of registered students, courses offered, and course enrollment details.

The system is developed using the C# programming language and follows the principles of Object-Oriented Programming (OOP). It uses a relational database to store data and provides a user-friendly graphical user interface.

This project aims to automate the manual process of student registration, thereby reducing errors, saving time, and improving efficiency.

SCOPE

The scope of the project encompasses the creation of a student registration form with basic input fields (e.g., name, age, email) and the handling of user inputs to exhibit their details.

Hardware Requirements:

Computer: A reasonably capable computer or laptop.

Processor: Dual-core processor or better for smooth development.

Memory (RAM): At least 4GB RAM for optimal performance while coding.

Storage: 128GB

Software Requirements:

Integrated Development Environment (IDE): Choose an IDE for C# development. Options include:

Visual Studio (Community edition is free and popular for C# development)

JetBrains Rider

SharpDevelop

Programming Language: C# Compiler, which is typically included in the chosen IDE.

Operating System: Windows is preferable due to better compatibility with Visual Studio, but C# can also be developed on Linux or macOS using alternative tools.

Database:

Microsoft SQL Server

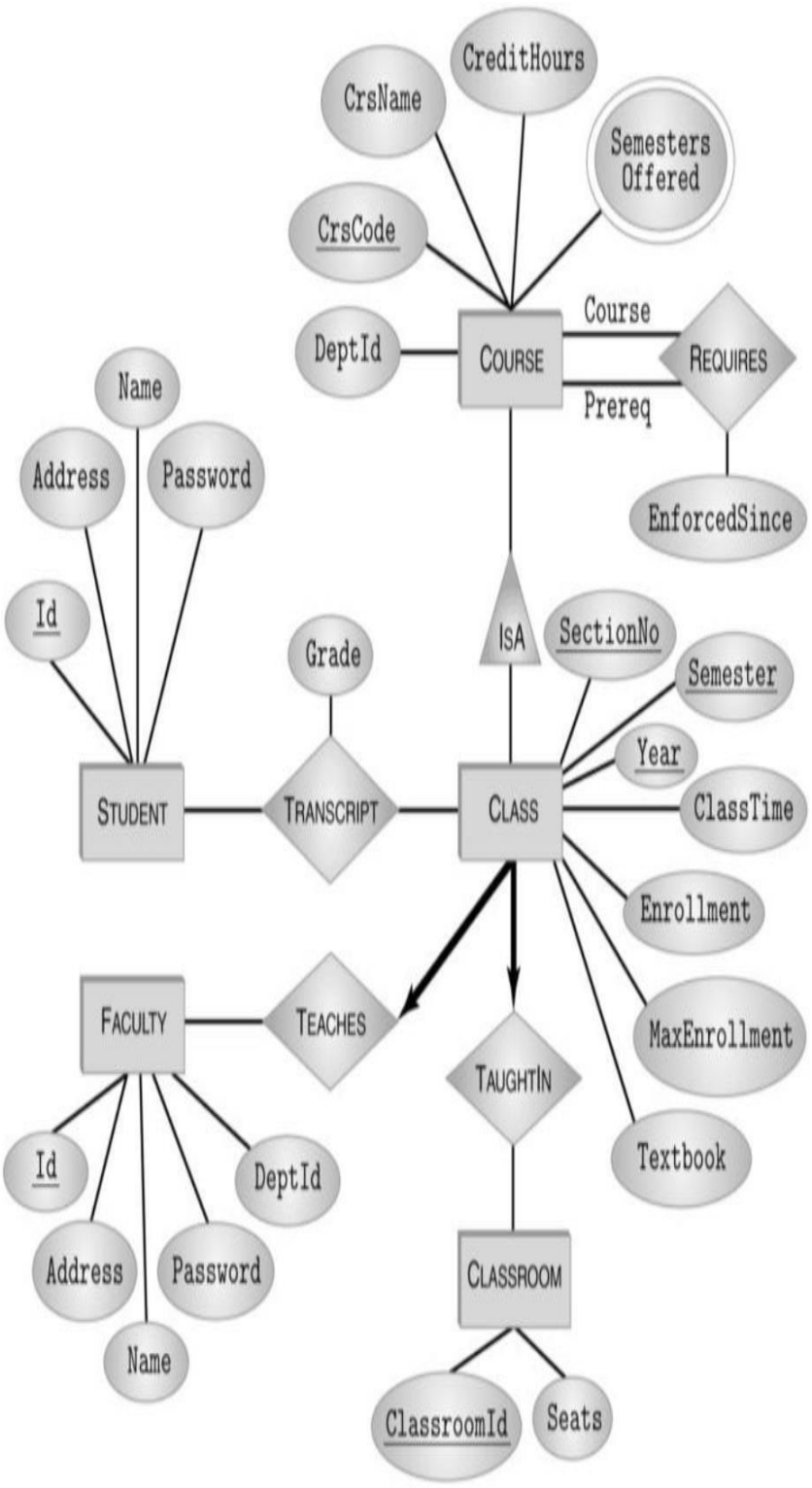
MySQL

SQLite

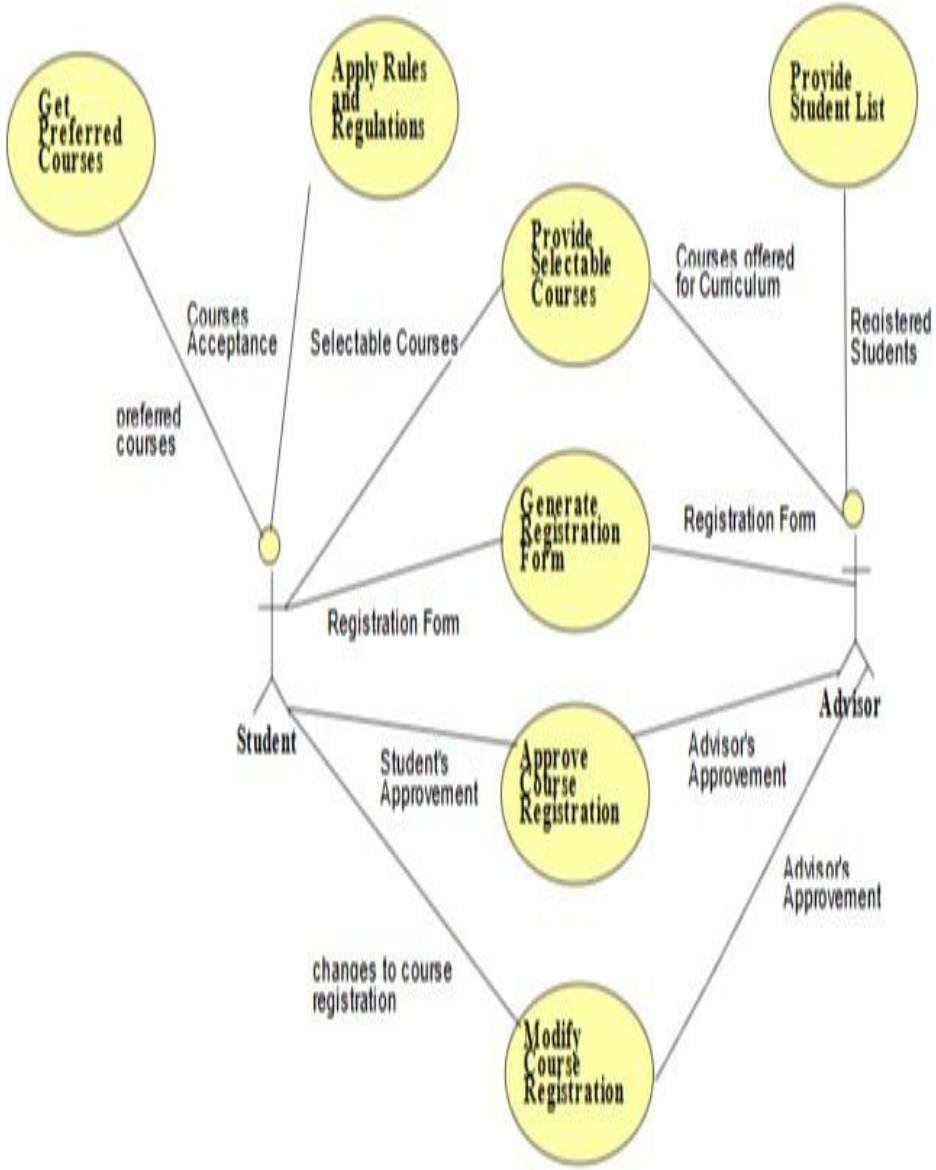
Frameworks or Libraries: Depending on the project's requirements, you might need specific frameworks or libraries for functionalities like data validation, user interface enhancements, etc.

Internet Connection: For accessing resources, updates, and potential online documentation.

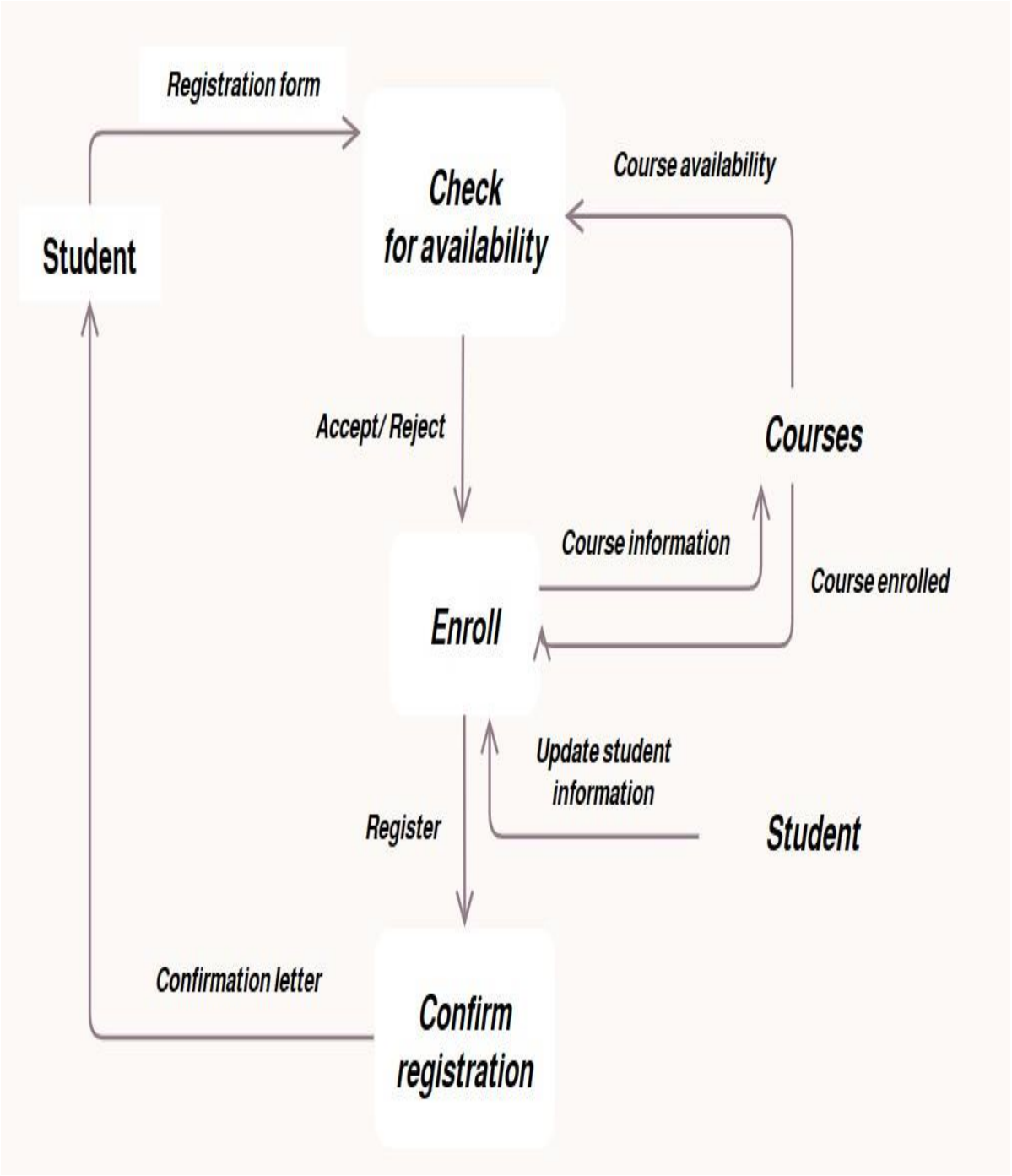
ER Diagram:


CreditHours
CrsName
Semesters Offered
CrsCode
DeptId
COURSE
Course
REQUIRES
Prereq
EnforcedSince
Name
Address
Password
Id
Grade
ISA
SectionNo
Semester
Year
ClassTime
STUDENT
TRANSCRIPT
CLASS
Enrollment
FACULTY
TEACHES
MaxEnrollment
TAUGHTIN
Textbook
Id
DeptId
Address
Password
Name
CLASSROOM
ClassroomId
Seats

Use Case Diagram:



DFD Diagram:


Registration form
Check for availability
Course availability
Student
Accept/ Reject
Courses
Course information
Course enrolled
Enroll
Update student information
Register
Student
Confirmation letter
Confirm registration

Basic Algorithm:

Initialize the student registration form:

Display the form with input fields for:

- Student name
- Age
- Gender
- Contact information
- Academic details (if needed)

When the user clicks the "**Submit**" button:

Validate the entered data:

- Ensure all required fields are filled.
- Validate the format of data (e.g., correct email format, numeric age, etc.)
- Perform any additional custom validation required.

If the data passes validation:

Create a Student object to store the entered information Store the Student

- If using a database:
  - Connect to the database.
  - Insert the Student object into the database.
- If using file storage:
  - Open the file.
  - Write the Student information to the file.

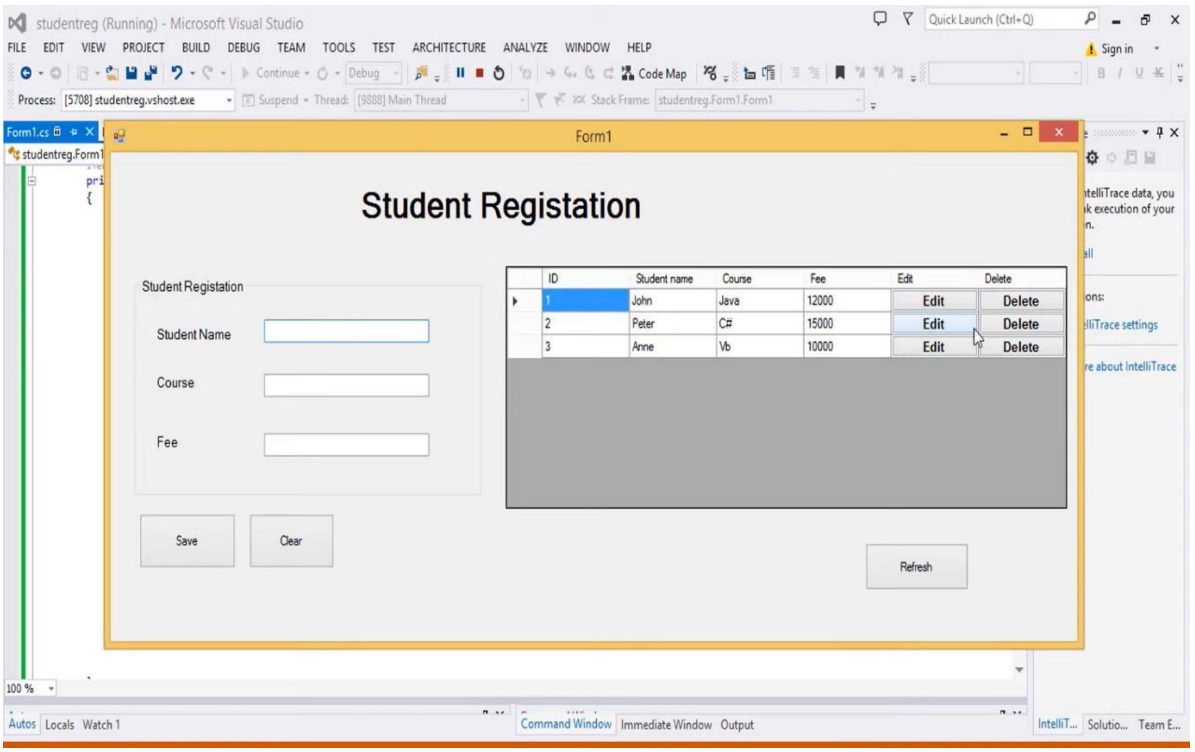
Display a success message indicating that the registration was successful.

If the data fails validation:

Display error messages indicating the validation issues.

Prompt the user to correct the errors.

Result:


studentreg (Running) - Microsoft Visual Studio
Form1
Student Registation
Student Registation
Student Name
Course
Fee
Save
Clear
Refresh
ID
Student name
Course
Fee
Edit
Delete
1
John
Java
12000
2
Peter
C#
15000
3
Anne
Vb
10000

Sample Code

```
using System.Data.SqlClient;
using System.Xml.Linq;

namespace WinFormsApp1
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
         LoadData();
      }
      SqlConnection con = new SqlConnection("Data Source=LAPTOP-K9FU71U8;Initial Catalog=Stu;;Integrated
Security=True;");
      SqlCommand cmd;
      SqlDataReader read;
      SqlDataAdapter drr;
      string id;
      bool Mode = true;
      string sql;

      public void LoadData()
      {
         try
         {
            sql = "select * from student";
            cmd = new SqlCommand(sql, con);
            con.Open();
            read = cmd.ExecuteReader();
            dataGridView1.Rows.Clear();

            while (read.Read())
            {
               dataGridView1.Rows.Add(read[0], read[1], read[2], read[3]);
            }
            con.Close();
         }
         catch (Exception ex)
         {
            MessageBox.Show(ex.Message);

         }
      }
      public void getID(String id)
      {
         sql = "select * from student where id = '" + id + "'  ";
         cmd = new SqlCommand(sql, con);
         con.Open();
         read = cmd.ExecuteReader();

         while (read.Read())
         {
            textBox2.Text = read[1].ToString();
            textBox1.Text = read[2].ToString();
            textBox3.Text = read[3].ToString();
         }
         con.Close();
      }

      private void label2_Click(object sender, EventArgs e)
```

```
{

}

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{

   if (e.ColumnIndex == dataGridView1.Columns["Edit"].Index && e.RowIndex >= 0)
   {
      Mode = false;
      id = dataGridView1.CurrentRow.Cells[0].Value.ToString();
      getID(id);
      EDIT.Text = "Edit";

   }
   else if (e.ColumnIndex == dataGridView1.Columns["Delete"].Index && e.RowIndex >= 0)
   {
      Mode = false;
      id = dataGridView1.CurrentRow.Cells[0].Value.ToString();
      sql = "delete from student where id  = @id ";
      con.Open();
      cmd = new SqlCommand(sql, con);
      cmd.Parameters.AddWithValue("@id ", id);
      cmd.ExecuteNonQuery();
      MessageBox.Show("Record Deleteeeee");
      con.Close();
   }

}

private void groupBox1_Enter(object sender, EventArgs e)
{

}

private void Form1_Load(object sender, EventArgs e)
{

}

private void button3_Click(object sender, EventArgs e)
{
   string name = textBox2.Text;
   string course = textBox1.Text;
   string fee = textBox3.Text;

   if (Mode == true)
   {
      sql = "insert into student(stname,course,fee) values(@stname,@course,@fee)";
      con.Open();
      cmd = new SqlCommand(sql, con);
      cmd.Parameters.AddWithValue("@stname", name);
      cmd.Parameters.AddWithValue("@course", course);
      cmd.Parameters.AddWithValue("@fee", fee);
      MessageBox.Show("Record Addddedddd");
      cmd.ExecuteNonQuery();

      textBox2.Clear();
      textBox1.Clear();
      textBox3.Clear();
      textBox2.Focus();

   }
```

```
            else
            {
                id = dataGridView1.CurrentRow.Cells[0].Value.ToString();
                sql = "update student set stname = @stname, course= @course,fee = @fee where id = @id";
                con.Open();
                cmd = new SqlCommand(sql, con);
                cmd.Parameters.AddWithValue("@stname",
                name);
                cmd.Parameters.AddWithValue("@course",
                course);
                cmd.Parameters.AddWithValue("@fee",
                fee); cmd.Parameters.AddWithValue("@id",
                id); MessageBox.Show("Record
                Updatedddddd"); cmd.ExecuteNonQuery();


                textBox2.Clear()
                ;
                textBox1.Clear()
                ;
                textBox3.Clear()
                ;
                textBox2.Focus(
                ); button3.Text
                = "Save"; Mode
                = true;

            }
            con.Close();

        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox2.Clear();
            textBox1.Clear();
            textBox3.Clear();
            textBox2.Focus();
            button3.Text =
            "Save"; Mode =
            true;
        }

        private void button4_Click(object sender, EventArgs e)
        {
            LoadData();

        }
    }
}
```

Conclusion:

"In summary, the Student Registration Form project has successfully streamlined the registration process, offering a user-friendly interface and prioritizing data security. This efficient system marks a significant improvement in administrative processes within educational institutions, enhancing accuracy and user experience. The collaborative effort of the development team and user feedback have contributed to the success of the project, making it a milestone in optimizing routine tasks for administrators and students."