Windows Forms Application – Project Report

Project Title: School Management System

Submitted by: Saif Ahamed UT010632

Introduction

This project is a basic Windows Forms Application created using C# and Visual Studio. It is designed to perform simple tasks like [adding records, saving data, displaying information, etc.].

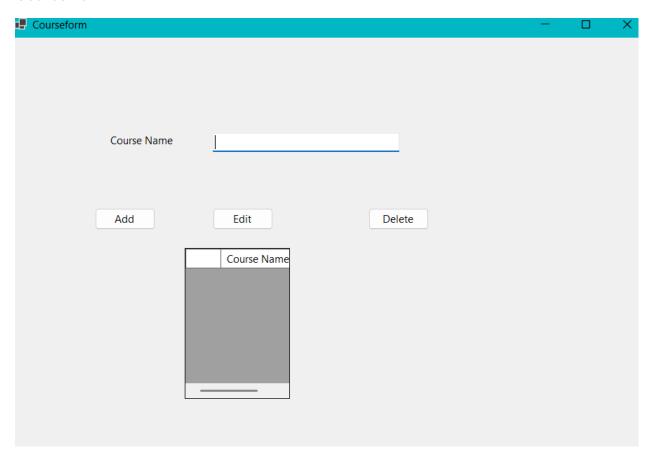| Tools & Technologies Used	Description
Tool Visual Studio	IDE used to develop the project
C#	Programming Language
.NET Framework	Platform for building the app
Windows Forms	For designing the user interface

Screenshots

Login Form

Hard-coded value (U. Name, Password)

Course Form

Courseform — ☐ ✕

Course Name [   ]

[Add]   [Edit]   [Delete]

	Course Name

Student Form

Studentform — ☐ ✕

Student Name [   ]

Address   [   ]

Age   [   ]

Gender   [Male   ∨]

[ADD]   [Edit]   [Delete]

	StudentId	Name	Age	Gender

Exam Form

Room Form

Marks Form

Markform — ☐ ✕

Student Id [   ]

Exam [▾]

Score [   ]

[Add]   [Edit]   [Delete]

Lecture Form

Lectureform — ☐ ✕

Lectrue Name [   ]

Adress [   ]

Age [   ]

[Add]   [Edit]   [Delete]

	LectureID	Name	Age

User Form

Codes Samples

```
1 reference | Saif ahmed, 3 days ago | 2 authors, 2 changes
public void AddStudent(Student student)
{
    using (var conn = datacon.GetConnection())
    {

        var cmd = conn.CreateCommand();
        cmd.CommandText = @"INSERT INTO Students (Name, Age, Address, Gender)
                            VALUES (@name, @age, @address, @gender)";
        cmd.Parameters.AddWithValue("@name", student.Name);
        cmd.Parameters.AddWithValue("@age", student.Age);
        cmd.Parameters.AddWithValue("@address", student.Address);
        cmd.Parameters.AddWithValue("@gender", student.Gender);
        cmd.ExecuteNonQuery();

    }
}
```

```csharp
public void UpdateStudent(Student student)
{
    using (var conn = datacon.GetConnection())
    {

        var cmd = conn.CreateCommand();
        cmd.CommandText = @"UPDATE Students SET
                            Name = @name,
                            Age = @age,
                            Address = @address,
                            Gender = @gender
                            WHERE StudentID = @id";
        cmd.Parameters.AddWithValue("@name", student.Name);
        cmd.Parameters.AddWithValue("@age", student.Age);
        cmd.Parameters.AddWithValue("@address", student.Address);
        cmd.Parameters.AddWithValue("@gender", student.Gender);
        cmd.Parameters.AddWithValue("@id", student.StudentId);
        cmd.ExecuteNonQuery();
    }
}
```

```csharp
public List<Student> GetAllStudents()
{
    List<Student> students = new List<Student>();

    using (var conn = datacon.GetConnection())
    {

        var cmd = conn.CreateCommand();
        cmd.CommandText = "SELECT * FROM Students";
        using (var reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                students.Add(new Student
                {
                    StudentId = reader["StudentID"] != DBNull.Value ? Convert.ToInt32(reader["StudentID"]) : 0,
                    Name = reader["Name"].ToString(),
                    Age = Convert.ToInt32(reader["Age"]),
                    Address = reader["Address"].ToString(),
                    Gender = reader["Gender"].ToString()
                });

            }
```

```csharp
private void button1_Click(object sender, EventArgs e)
{
    string username = txtname.Text.Trim();
    string password = txtpassword.Text.Trim();

    if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(password))
    {
        MessageBox.Show("Please enter both username and password.");
        return;
    }

    User user = loginController.Authenticate(username, password);

    if (user != null)
    {
        //MessageBox.Show($"Welcome {user.UserName}!", "Login Successful", MessageBoxButtons.OK, MessageBoxIcon.Information);

        // Hide this form and show the main form
        this.Hide();
        Mainform MainForm = new Mainform(); // or pass user info if needed
        MainForm.Show();
    }
    else
    {
```

```csharp
private void btnexams_Click(object sender, EventArgs e)
{

    Exam exam = new Exam();
    exam.Show();
}
```

This method was edited by 1 author in 3 unique commits.
(Last 1000 changes in the past 12 months) (Alt+8)

1 reference | Saif ahmed, 5 hours ago | 1 author, 3 changes

```csharp
private void btnlogout_Click(object sender, EventArgs e)
{

    DialogResult result = MessageBox.Show("Are you sure you want to log out?", "Confirm Logout", MessageBoxButtons.YesNo, MessageBoxIc

    if (result == DialogResult.Yes)
    {
        this.Close();
        Login loginForm = new Login();
        loginForm.Show();
    }
}
```

1 reference | Saif ahmed, 5 hours ago | 1 author, 2 changes

1 reference | Saif ahmed, 3 days ago | 1 author, 1 change
```csharp
private void LoadGenderOptions()
{
    cmbgender.Items.Clear();
    cmbgender.Items.Add("Male");
    cmbgender.Items.Add("Female");
    cmbgender.Items.Add("Other");
    cmbgender.SelectedIndex = 0;
}
```
4 references | Saif ahmed, 3 days ago | 1 author, 1 change
```csharp
private void LoadStudents()
{
    dgvstudent.DataSource = controller.GetAllStudents();
}
```

1 reference | Saif ahmed, 3 days ago | 1 author, 1 change
```csharp
private void btnadd_Click(object sender, EventArgs e)
{
    Student student = new Student
    {
        Name = txtstudentname.Text,
        Address = txtaddress.Text,
        Age = int.TryParse(txtage.Text, out int age) ? age : 0,
        Gender = cmbgender.SelectedItem.ToString()
    };
```