1.

public class ButtonInfo

{

public string Name { get; set; }

public string Text { get; set; }

public int X { get; set; }

public int Y { get; set; }

public int Width { get; set; }

public int Height { get; set; }

public string ColorCode { get; set; }

}

public static List<ButtonInfo> GetButtonInfo()

{

List<ButtonInfo> buttonList = new List<ButtonInfo>();

// TODO: Retrieve button information from the database and add it to the list

return buttonList;

}

private void MainForm\_Load(object sender, EventArgs e)

{

List<ButtonInfo> buttonList = GetButtonInfo();

foreach (ButtonInfo buttonInfo in buttonList)

{

Button button = new Button();

button.Name = buttonInfo.Name;

button.Text = buttonInfo.Text;

button.Location = new Point(buttonInfo.X, buttonInfo.Y);

button.Size = new Size(buttonInfo.Width, buttonInfo.Height);

button.BackColor = ColorTranslator.FromHtml(buttonInfo.ColorCode);

flowLayoutPanel1.Controls.Add(button);

}

}

private void addButton\_Click(object sender, EventArgs e)

{

List<ButtonInfo> buttonList = GetButtonInfo();

foreach (ButtonInfo buttonInfo in buttonList)

{

Button button = new Button();

button.Name = buttonInfo.Name;

button.Text = buttonInfo.Text;

button.Location = new Point(buttonInfo.X, buttonInfo.Y);

button.Size = new Size(buttonInfo.Width, buttonInfo.Height);

button.BackColor = ColorTranslator.FromHtml(buttonInfo.ColorCode);

flowLayoutPanel1.Controls.Add(button);

}

}

2.

private void cancelButton\_Click(object sender, EventArgs e)

{

ClearStudentDetailControls();

dataGridView1.ClearSelection();

}

private void refreshButton\_Click(object sender, EventArgs e)

{

ClearStudentDetailControls();

using var db = new MyDbContext();

List<Student> students = db.Students.ToList();

dataGridView1.DataSource = students;

}

private void editButton\_Click(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

DataGridViewRow selectedRow = dataGridView1.SelectedRows[0];

int id = int.Parse(idTextBox.Text);

using var db = new MyDbContext();

Student student = db.Students.Find(id);

student.Name = nameTextBox.Text;

student.Age = int.Parse(ageTextBox.Text);

student.Address = addressTextBox.Text;

db.SaveChanges();

dataGridView1.DataSource = db.Students.ToList();

}

}

private void dataGridView1\_SelectionChanged(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

DataGridViewRow selectedRow = dataGridView1.SelectedRows[0];

Student student = (Student)selectedRow.DataBoundItem;

idTextBox.Text = student.Id.ToString();

nameTextBox.Text = student.Name;

ageTextBox.Text = student.Age.ToString();

addressTextBox.Text = student.Address;

idTextBox.Enabled = false;

}

else

{

ClearStudentDetailControls();

}

}

private void ClearStudentDetailControls()

{

idTextBox.Text = "";

nameTextBox.Text = "";

ageTextBox.Text = "";

addressTextBox.Text = "";

idTextBox.Enabled = true;

}

private void addButton\_Click(object sender, EventArgs e)

{

Student student = new Student

{

Name = nameTextBox.Text,

Age = int.Parse(ageTextBox.Text),

Address = addressTextBox.Text

};

using var db = new MyDbContext();

db.Students.Add(student);

db.SaveChanges();

dataGridView1.DataSource = db.Students.ToList();

}

private void MainForm\_Load(object sender, EventArgs e)

{

using var db = new MyDbContext();

List<Student> students = db.Students.ToList();

dataGridView1.DataSource = students;

}

public class Student

{

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public string Address { get; set; }

}

public class MyDbContext : DbContext

{

public DbSet<Student> Students { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

// TODO: Configure the database connection string

optionsBuilder.UseSqlServer("<connection-string>");

}

}

3.

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<AppDbContext>(options =>

options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

services.AddRazorPages();

}

public class IndexModel : PageModel

{

private readonly AppDbContext \_dbContext;

public IndexModel(AppDbContext dbContext)

{

\_dbContext = dbContext;

}

public IList<Student> Students { get; set; }

[BindProperty(SupportsGet = true)]

public string SelectedSex { get; set; }

public SelectList Genders { get; } = new SelectList(new[]

{

new { Value = "Male", Text = "Male" },

new { Value = "Female", Text = "Female" }

}, "Value", "Text");

public void OnGet()

{

Students = \_dbContext.Students.ToList();

}

public void OnPost()

{

if (!string.IsNullOrEmpty(SelectedSex))

{

Students = \_dbContext.Students.Where(s => s.Sex == SelectedSex).ToList();

}

else

{

Students = \_dbContext.Students.ToList();

}

}

}

@page

@model IndexModel

@{

ViewData["Title"] = "Students";

}

<h2>@ViewData["Title"]</h2>

<form method="post">

<select asp-for="SelectedSex" asp-items="@Model.Genders"></select>

<button type="submit">Filter</button>

</form>

<table class="table">

<thead>

<tr>

<th>ID</th>

<th>Full Name</th>

<th>Sex</th>

<th>DOB</th>

<th>Email</th>

</tr>

</thead>

<tbody>

@foreach (var student in Model.Students)

{

<tr>

<td>@student.Id</td>

<td>@student.FullName</td>

<td>@student.Sex</td>

<td>@student.Dob.ToShortDateString()</td>

<td>@student.Email</td>

</tr>

}

</tbody>

</table>

Add-Migration InitialCreate

Update-Database

public class AppDbContext : DbContext

{

public AppDbContext(DbContextOptions<AppDbContext> options)

: base(options)

{

}

public DbSet<Student> Students { get; set; }

}

public class Student

{

public int Id { get; set; }

public string FullName { get; set; }

public string Sex { get; set; }

public DateTime Dob { get; set; }

public string Email { get; set; }

}

Install-Package Microsoft.EntityFrameworkCore.SqlServer