Appsettings.json

```
"AllowedHosts": "*",
  "ConnectionStrings": {
    "Dev_Connection": "server=localhost;user id=root;password=;database=deerwalk;"
  }
```

Program.cs

```
builder.Services.AddDbContext<DataContext>(option =>
{
    option.UseMySQL(builder.Configuration.GetConnectionString("Dev_Connection"));
});

var app = builder.Build();
```

DataContext.cs

```
using Microsoft.EntityFrameworkCore;

namespace prjAspCore_SecA.Model
{
    public class DataContext : DbContext
    {
        public DataContext(DbContextOptions options) : base(options)
        {

        }
        public DbSet<PhonebookInfo> phonebook { get; set; }

    }
}
```

PhonebookInfo.cs

```
using System.ComponentModel.DataAnnotations;

namespace prjAspCore_SecA.Model
{
    public class PhonebookInfo
    {
        [Key]
        public int id { get; set; }
        public string firstname { get; set; }
        public string lastname { get; set; }
        public string middlename { get; set; }
        public string title { get; set; }
        public string gender { get; set; }
        public string contact { get; set; }
        public string photo { get; set; }
    }
}
```

Create.cshtml

```
@page
@model prjAspCore_SecA.Pages.Phonebook.CreateModel

@{
    ViewData["Title"] = "Create";
}

<h1>Create</h1>

<h4>PhonebookInfo</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form method="post" enctype="multipart/form-data">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="PhonebookInfo.firstname" class="control-label"></label>
                <input asp-for="PhonebookInfo.firstname" class="form-control" />
                <span asp-validation-for="PhonebookInfo.firstname" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="PhonebookInfo.lastname" class="control-label"></label>
                <input asp-for="PhonebookInfo.lastname" class="form-control" />
                <span asp-validation-for="PhonebookInfo.lastname" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="PhonebookInfo.middlename" class="control-label"></label>
                <input asp-for="PhonebookInfo.middlename" class="form-control" />
                <span asp-validation-for="PhonebookInfo.middlename" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="PhonebookInfo.title" class="control-label"></label>
                <select asp-for="PhonebookInfo.title" asp-items="Model.Titles" >
                    <option value="">Please select title</option>
                </select>
                <span asp-validation-for="PhonebookInfo.title" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="PhonebookInfo.gender" class="control-label"></label>
               @foreach (var gender in Model.Genders)
                {
                    <input type="radio" asp-for="PhonebookInfo.gender" value="@gender" />@gender<br />
                }
    <span asp-validation-for="PhonebookInfo.gender" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="PhonebookInfo.contact" class="control-label"></label>
                <input asp-for="PhonebookInfo.contact" class="form-control" />
                <span asp-validation-for="PhonebookInfo.contact" class="text-danger"></span>
            </div>
            <div class="form-group">

                <label asp-for="UploadedFile" class="control-label"></label>
               <input asp-for="UploadedFile" type="file">
                <span asp-validation-for="UploadedFile" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>

<div>
    <a asp-page="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

Create.cshtml.cs

```
private readonly prjAspCore_SecA.Model.DataContext _context;

public CreateModel(prjAspCore_SecA.Model.DataContext context)
        {
            _context = context;
        }

[BindProperty]
public PhonebookInfo PhonebookInfo { get; set; } = default!;
//using asynchronous programming
public async Task<IActionResult> OnPostAsync()
        {
                    if (!ModelState.IsValid)
                    {
                return Page();
                    }
                    _context.Phonebook.Add(PhonebookInfo);
            await _context.SaveChangesAsync();
            return RedirectToPage("./Index");
        }
```

OR

```
public IActionResult OnPost()
        {
                     _context.Phonebook.Add(PhonebookInfo);
                     _context.SaveChanges();

                     return RedirectToPage("./Index");
        }
```

Complete Code

```
namespace prjAspCore_SecA.Pages.Phonebook
{
    public class TitleModel
    {
        [BindProperty]
        public int id { get; set; }

        [BindProperty]
        public string name { get; set; }
    }
    public class CreateModel : PageModel
    {
        public string[] Genders = new[] { "Male", "Female", "Unspecified" };

        private readonly prjAspCore_SecA.Model.DataContext _context;
        private readonly IHostEnvironment _environment;
        public CreateModel(prjAspCore_SecA.Model.DataContext context, IHostEnvironment environment)
        {
            _context = context;
            _environment = environment;
        }
        public SelectList Titles { get; set; }
        public IActionResult OnGet()
        {
            /*  Dictionary<int, string> dic = new Dictionary<int, string>();
              dic.Add(1, "Mr");
              dic.Add(2, "Mrs");*/
            List<TitleModel> dic = new List<TitleModel>();// { "Mr", "Mrs" };
            TitleModel t1 = new TitleModel();
            t1.id = 1; t1.name = "Mr";
            dic.Add(t1);
            TitleModel t2 = new TitleModel();
            t2.id = 1; t2.name = "Mrs";
            dic.Add(t2);

            this.Titles = new SelectList(dic, "id", "name");
            return Page();
        }

        [BindProperty]
        public PhonebookInfo PhonebookInfo { get; set; } = default!;

        public IFormFile UploadedFile { get; set; }

        // To protect from overposting attacks, see https://aka.ms/RazorPagesCRUD
        public IActionResult OnPost()
        {
            PhonebookInfo.photo = UploadedFile.FileName;
            //if (!ModelState.IsValid || _context.Phonebook == null || PhonebookInfo == null)
            //{
            //    ModelState.Clear();
            //    return Page();
            //}
            string fullpath = System.AppDomain.CurrentDomain.BaseDirectory.ToString();
            var path = Path.Combine(_environment.ContentRootPath, "images", UploadedFile.FileName);
            var stream = new FileStream(path, FileMode.Create);
            UploadedFile.CopyToAsync(stream);
            _context.phonebook.Add(PhonebookInfo);
            _context.SaveChanges();

            return RedirectToPage("Create");
        }
    }
}
```

List.cshtml

```
@page
@model prjAspCore_SecA.Pages.Phonebook.ListModel

@{
    ViewData["Title"] = "List";
}

<h1>List</h1>

<p>
    <a asp-page="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.PhonebookInfo[0].firstname)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.PhonebookInfo[0].lastname)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.PhonebookInfo[0].middlename)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.PhonebookInfo[0].title)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.PhonebookInfo[0].gender)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.PhonebookInfo[0].contact)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
@foreach (var item in Model.PhonebookInfo) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.firstname)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.lastname)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.middlename)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.title)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.gender)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.contact)
            </td>
            <td>
                <a asp-page="./Edit" asp-route-id="@item.id">Edit</a> |
                <a asp-page="./Details" asp-route-id="@item.id">Details</a> |
                <a asp-page="./Delete" asp-route-id="@item.id">Delete</a>
            </td>
        </tr>
}
    </tbody>
</table>
```

ListModel.cshtml.cs

```
namespace prjAspCore_SecA.Pages.Phonebook
{
    public class ListModel : PageModel
    {
        private readonly prjAspCore_SecA.Model.DataContext _context;

        public ListModel(prjAspCore_SecA.Model.DataContext context)
        {
            _context = context;
        }

        public IList<PhonebookInfo> PhonebookInfo { get;set; } = default!;

        public async Task OnGetAsync()
        {
            if (_context.phonebook != null)
            {
                PhonebookInfo = await _context.phonebook.ToListAsync();
            }
        }
    }
}
```

Delete.cshtml

```
@page
@model prjAspCore_SecB.Pages.All.DeleteModel

@{
    ViewData["Title"] = "Delete";
}

<h1>Delete</h1>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>StudentInfo</h4>
    <hr />
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.StudentInfo.firstname)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.StudentInfo.firstname)
        </dd>
    </dl>

    <form method="post">
        <input type="hidden" asp-for="StudentInfo.id" />
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a asp-page="./Index">Back to List</a>
    </form>
</div>

namespace prjAspCore_SecB.Pages.All
{
    public class DeleteModel : PageModel
    {
        private readonly prjAspCore_SecB.Model.DataContext _context;

        public DeleteModel(prjAspCore_SecB.Model.DataContext context)
        {
            _context = context;
        }

        [BindProperty]
        public StudentInfo StudentInfo { get; set; } = default!;

        public void OnGet(int? id)
        {
            var studentinfo = _context.student.Find(id);
            StudentInfo = studentinfo;

        }

        public IActionResult OnPost(int? id)
        {
            StudentInfo studentinfo = _context.student.Find(id);

            if (studentinfo != null)
            {
                StudentInfo = studentinfo;
                _context.student.Remove(StudentInfo);
                _context.SaveChanges();
            }
            return Page();
        }
    }
}
```

Edit.cshtml

```
@page
@model prjAspCore_SecB.Pages.All.EditModel

@{
    ViewData["Title"] = "Edit";
}

<h1>Edit</h1>

<h4>StudentInfo</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form method="post">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <input type="hidden" asp-for="StudentInfo.id" />
            <div class="form-group">
                <label asp-for="StudentInfo.firstname" class="control-label"></label>
                <input asp-for="StudentInfo.firstname" class="form-control" />
                <span asp-validation-for="StudentInfo.firstname" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Save" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>


namespace prjAspCore_SecB.Pages.All
{
    public class EditModel : PageModel
    {
        private readonly prjAspCore_SecB.Model.DataContext _context;

        public EditModel(prjAspCore_SecB.Model.DataContext context)
        {
            _context = context;
        }

        [BindProperty]
        public StudentInfo studentinfo { get; set; } = default!;

        public IActionResult OnGet(int? id)
        {
            studentinfo = _context.student.Find(id);
            return Page();
        }

        // To protect from overposting attacks, enable the specific properties you want to bind to.
        // For more details, see https://aka.ms/RazorPagesCRUD.
        public async Task<IActionResult> OnPostAsync()
        {
            _context.Attach(StudentInfo).State = EntityState.Modified;
            await _context.SaveChangesAsync();
            return RedirectToPage("./Index");
        }

    }
}
```