Qn 1:

public class Product

{

[Key]

public int Id { get; set; }

public string Name { get; set; }

public string Category { get; set; }

[Range(1, 50, ErrorMessage = "Invalid price")]

public double Price { get; set; }

[Range(1, 20, ErrorMessage = "Invalid quantity")]

public int Quantity { get; set; }

public string Pic { get; set; }

public string Description { get; set; }

}

public class ShopContext : DbContext

{

public ShopContext(DbContextOptions options) : base(options)

{

}

public DbSet<Product> Products { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder) //override onmodel tab

{

modelBuilder.Entity<Product>().HasData(

new Product() { Id = 101, Name = "Green Peppers", Category = "Food", Price=1, Quantity=20, Pic= "/images/green.png", Description="From Malaysia" }

);

}

}

public interface IRepo<K, T>

{

ICollection<T> GetAll();

T Get(K k);

T Add(T item);

bool Remove(K id);

bool Update(T item);

}

public class ProductRepo : IRepo<int, Product>

{

private readonly ShopContext \_context;

public ProductRepo(ShopContext context)

{

\_context = context;

}

public Product Add(Product item)

{

\_context.Products.Add(item);

\_context.SaveChanges();

return item;

}

public Product Get(int id)

{

Product product = \_context.Products.FirstOrDefault(x => x.Id == id);

return product;

}

public ICollection<Product> GetAll()

{

return \_context.Products.ToList();

}

public bool Remove(int id)

{

Product product = Get(id);

\_context.Products.Remove(product);

\_context.SaveChanges();

return true;

}

public bool Update(Product item)

{

Product product = \_context.Products.FirstOrDefault(x => x.Id == item.Id);

if (product != null)

{

product.Name = item.Name;

product.Category = item.Category;

product.Price = item.Price;

product.Quantity = item.Quantity;

product.Pic = item.Pic;

product.Description = item.Description;

\_context.Products.Update(product);

\_context.SaveChanges();

return true;

}

return false;

}

}

public class ProductController : Controller

{

private readonly IRepo<int, Product> \_repo;

public ProductController(IRepo<int, Product> repo)

{

\_repo = repo;

}

public IActionResult Index()

{

return View(\_repo.GetAll());

}

public IActionResult Details(int id)

{

Product product = \_repo.Get(id);

return View(product);

}

public IActionResult Create()

{

ViewBag.Category = GetProductCategories();

ViewBag.Pic = GetProductPic();

return View(new Product());

}

[HttpPost]

public IActionResult Create(Product product)

{

//server side also can do validations

if (ModelState.IsValid)

{

\_repo.Add(product);

return RedirectToAction("Index");

}

return RedirectToAction("Create");

}

public IActionResult Edit(int id)

{

Product product = \_repo.Get(id);

ViewBag.Category = GetProductCategories();

ViewBag.Pic = GetProductPic();

return View(product);

}

[HttpPost]

public IActionResult Edit(int id, Product product)

{

\_repo.Update(product);

return RedirectToAction("Index");

}

public IActionResult Delete(int id)

{

Product product = \_repo.Get(id);

return View(product);

}

[HttpPost]

public IActionResult Delete(int id, Product customer)

{

\_repo.Remove(id);

return RedirectToAction("Index");

}

[HttpPost]

public IActionResult QtyOfProduct(int qty)

{

int myValue = qty;

return RedirectToAction("Index");

}

IEnumerable<SelectListItem> GetProductCategories()

{

List<SelectListItem> category = new List<SelectListItem>();

category.Add(new SelectListItem { Text = "Food", Value = "Food" });

category.Add(new SelectListItem { Text = "Toy", Value = "Toy" });

category.Add(new SelectListItem { Text = "Clothing", Value = "Clothing" });

return category;

}

IEnumerable<SelectListItem> GetProductPic()

{

List<SelectListItem> pics = new List<SelectListItem>();

pics.Add(new SelectListItem { Text = "Green Bell Pepper", Value = "/images/green.png" });

pics.Add(new SelectListItem { Text = "Red Bell Pepper", Value = "/images/red.png" });

pics.Add(new SelectListItem { Text = "Stacking Wooden Hoop", Value = "/images/teddy.png" });

return pics;

}

}

@model IEnumerable<PassingInputAssignmentApp.Models.Product>

@{

ViewData["Title"] = "Index";

}

<h1>Index</h1>

<p>

<**a** **asp-action**="Create">Create New</**a**>

</p>

<style>

#divCard{

position:relative;

float:left;

height:20%;

width:35%;

}

</style>

@foreach (var item in Model)

{

<div id="divCard" class="card">

<!--Card image-->

<div class="view">

<img width="150" height="150" src="@Url.Content(item.Pic)" alt="No Pic" class="card-img-top" alt="photo">

<a href="#">

<div class="mask rgba-white-slight"></div>

</a>

</div>

<!--Card content-->

<div class="card-body">

<!--Title-->

<h4 class="card-title">@item.Name</h4>

<!--Text-->

Category: <label class="card-text">@item.Category</label>

<p class="card-text">@item.Description</p>

<a href="/Product/Details/@item.Id" class="btn btn-primary">Buy 1 for $@item.Price</a>

@\*<a href="@Html.ActionLink("Show more", "Details",new{id=@item.Id})">Buy for $@item.Price</a>\*@

</div>

</div>

}

@model PassingInputAssignmentApp.Models.Product

@{

ViewData["Title"] = "Create";

}

<h1>Create</h1>

<h4>Product</h4>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="Create">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

@\*<div class="form-group">

<label asp-for="Id" class="control-label"></label>

<input asp-for="Id" class="form-control" />

<span asp-validation-for="Id" class="text-danger"></span>

</div>\*@

<div class="form-group">

<**label** **asp-for**="Name" class="control-label"></**label**>

<**input** **asp-for**="Name" class="form-control" />

<**span** **asp-validation-for**="Name" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Category" class="control-label"></**label**>

@\*<input asp-for="Category" class="form-control" />\*@

<**select** **asp-for**="Category" **asp-items**="@ViewBag.Category" class="form-control">

<**option**>Please select a Category</**option**>

</**select**>

<**span** **asp-validation-for**="Category" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Price" class="control-label"></**label**>

<**input** **asp-for**="Price" class="form-control" />

<**span** **asp-validation-for**="Price" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Quantity" class="control-label"></**label**>

<**input** **asp-for**="Quantity" class="form-control" />

<**span** **asp-validation-for**="Quantity" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Pic" class="control-label"></**label**>

@\*<input asp-for="Pic" class="form-control" />\*@

<**select** **asp-for**="Pic" **asp-items**="@ViewBag.Pic" class="form-control">

<**option**>Please select a Picture</**option**>

</**select**>

<**span** **asp-validation-for**="Pic" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Description" class="control-label"></**label**>

<**input** **asp-for**="Description" class="form-control" />

<**span** **asp-validation-for**="Description" 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-action**="Index">Back to List</**a**>

</div>

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

@model PassingInputAssignmentApp.Models.Product

@{

ViewData["Title"] = "Details";

}

<h1>Details</h1>

<div>

<h4>Product</h4>

<hr />

<dl class="row">

<**form** **asp-action**="QtyOfProduct">

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Id)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Id)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Name)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Name)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Category)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Category)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Price)

</dt>

<dd class = "col-sm-10">

$@Html.DisplayFor(model => model.Price)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Quantity)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Quantity)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Pic)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Pic)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Description)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Description)

</dd>

<dt class = "col-sm-2">

@Html.Label("Add Qty to Cart")

</dt>

<dd class = "col-sm-10">

@\*@Html.TextBoxFor("Qty", new { @type = "number" })\*@

<input type="number" name="Qty" />

@\*@Html.TextBox("sub\_button","Full Name",new{type="submit"})\*@

<input type="submit" value="Qty" class="btn btn-primary" />

</dd>

</**form**>

</dl>

</div>

<div>

@Html.ActionLink("Edit", "Edit", new { id = Model.Id }) |

<**a** **asp-action**="Index">Back to List</**a**>

</div>

--------



