06 Update Delete Product

Goal
Allow admin to edit and delete products.
This usually earns marks because it shows CRUD.

CRUD meaning
Create, Read, Update, Delete

Step 1. Add Edit actions in controller

```
[HttpGet]
public IActionResult Edit(int Id)
{
    var gEProduct = context.GEProducts.Find(Id);
    if (gEProduct == null)
    {
        return RedirectToAction("Index", "Product");
    }

    var gEProductDTO = new GEProductDTO
    {
        Name = gEProduct.Name,
        Brand = gEProduct.Brand,
        Category = gEProduct.Category,
        Description = gEProduct.Description,
        Price = gEProduct.Price,
    };

    ViewData["ImageFileName"] = gEProduct.ImageFileName;


    return View(gEProductDTO);
}

[HttpPost]
public IActionResult Edit(int Id, GEProductDTO gEProductDTO)
{
    var gEProduct = context.GEProducts.Find(Id);
    if (gEProduct == null)
    {
        return RedirectToAction("Index", "Product");
    }
    if (!ModelState.IsValid)
    {
        ViewData["ImageFileName"] = gEProduct.ImageFileName;
        return View(gEProductDTO);
    }
    if (gEProductDTO.ImageFile != null)
    {
        string folderpath = Path.Combine(webHostEnvironment.WebRootPath, "geproducts");
        string fullFilePath = Path.Combine(folderpath, gEProductDTO.ImageFile.FileName);
        using var stream = new FileStream(fullFilePath, FileMode.Create);
        gEProductDTO.ImageFile.CopyTo(stream);
        gEProduct.ImageFileName = gEProductDTO.ImageFile.FileName;
    }
    gEProduct.Name = gEProductDTO.Name;
    gEProduct.Brand = gEProductDTO.Brand;
    gEProduct.Category = gEProductDTO.Category;
    gEProduct.Description = gEProductDTO.Description;
    gEProduct.Price = gEProductDTO.Price;
    context.GEProducts.Update(gEProduct);
    context.SaveChanges();
    return RedirectToAction("Index", "Product");

}

```

Explanation

GET Edit loads existing product and pre fills the form.

POST Edit saves changes.

If a new image is uploaded, it overwrites ImageFileName.

Step 2. Create Edit view

SAME WAS AS OTHERS RIGHT CLICK ON VIEW IN EDIT METHOD THEN CLICK ON ADD VIEW THEN GIVE IT THE SAME NAME Edit

```
@model GEProductDTO

@{
    var imagefilename = ViewData["ImageFileName"] as string;
}

<div class="container py-4" style="max-width: 900px;">
    <div class="d-flex align-items-center justify-content-between flex-wrap gap-2 mb-3">
        <h2 class="m-0">Edit Product</h2>
        <a class="btn btn-outline-secondary" asp-controller="Product" asp-action="Index">Back</a>
    </div>

    <div class="card border-info shadow-sm mb-3">
        <div class="card-body">
            <div class="d-flex gap-3 align-items-start flex-wrap">
                <div class="bg-light rounded" style="width: 140px; height: 105px; overflow: 
hidden;">
                    @if (!string.IsNullOrWhiteSpace(imagefilename))
                    {
                        <img src="@Url.Content($"~/geproducts/{imagefilename}")"
                             alt="@Model.Name"
                             style="width: 100%; height: 100%; object-fit: cover;" />
                    }
                    else
                    {
                        <div class="d-flex align-items-center justify-content-center h-100 text-
muted">
                            No image
                        </div>
                    }
                </div>

                <div class="flex-grow-1">
                    <div class="d-flex align-items-start justify-content-between gap-2 flex-wrap">
```

```
                        <div>
                            <h5 class="mb-1">@Model.Name</h5>
                            <div class="text-muted small">@Model.Brand</div>
                        </div>
                        <div class="text-end">
                            <div class="fw-semibold">@Model.Price.ToString("C")</div>
                            <span class="badge bg-secondary">@Model.Category</span>
                        </div>
                    </div>
                    <p class="text-muted small mt-2 mb-0">
                        @Model.Description
                    </p>
                </div>
            </div>
        </div>
    </div>

    <div class="card shadow-sm">
        <div class="card-body">
            <form asp-controller="Product"
                  asp-action="Edit"
                  method="post"
                  enctype="multipart/form-data">

                <div class="row g-3">
                    <div class="col-md-6">
                        <label asp-for="Name" class="form-label"></label>
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-danger"></span>
                    </div>

                    <div class="col-md-6">
                        <label asp-for="Brand" class="form-label"></label>
                        <input asp-for="Brand" class="form-control" />
                        <span asp-validation-for="Brand" class="text-danger"></span>
                    </div>
                    <div class="col-md-6">
                        <label asp-for="Category" class="form-label"></label>
                        <select asp-for="Category" class="form-select">
                            <option value="">Select category</option>
                            <option value="Solar Panels">Solar Panels</option>
                            <option value="Energy Storage">Energy Storage</option>
                            <option value="EV Charging">EV Charging</option>
                            <option value="Inverters">Inverters</option>
                            <option value="Electric Mobility">Electric Mobility</option>
                        </select>
                        <span asp-validation-for="Category" class="text-danger"></span>
                    </div>

                    <div class="col-md-6">
                        <label asp-for="Price" class="form-label"></label>
                        <input asp-for="Price" type="number" step="0.01" class="form-control" />
                        <span asp-validation-for="Price" class="text-danger"></span>
                    </div>

                    <div class="col-12">
                        <label asp-for="ImageFile" class="form-label">Replace Image</label>
                        <input asp-for="ImageFile" type="file" class="form-control" />
                        <span asp-validation-for="ImageFile" class="text-danger"></span>

                        <div class="form-text">
                            Leave empty to keep the current image.
                        </div>
                    </div>

                    <div class="col-12">
                        <label asp-for="Description" class="form-label"></label>
                        <textarea asp-for="Description"
                                  class="form-control"
                                  rows="5"></textarea>
                        <span asp-validation-for="Description" class="text-danger"></span>
                    </div>

                </div>

                <div class="d-flex gap-2 mt-4">
                    <button type="submit" class="btn btn-primary">Save changes</button>
                    <a class="btn btn-outline-secondary" asp-controller="Product" asp-
action="Index">Cancel</a>
                </div>
            </form>
        </div>
    </div>
</div>

```

Step 3. Add Delete actions

```
    [HttpGet]
    public IActionResult Delete(int Id)
    {
        var gEProduct = context.GEProducts.Find(Id);
        if (gEProduct == null)
        {
            return RedirectToAction("Index", "Product");
        }

        string imageFullPath = webHostEnvironment.WebRootPath + "/geproducts/" + gEProduct;

        if (System.IO.File.Exists(imageFullPath))
            System.IO.File.Delete(imageFullPath);

        context.GEProducts.Remove(gEProduct);
        context.SaveChanges();
        return RedirectToAction("Index");

    }

```

Explanation

Delete uses POST so it is harder to trigger by accident.

Remove deletes the row, SaveChanges commits it.

Step 4. Add Edit and Delete buttons on Index page

```
Update the table in Index.cshtml to include actions.
<th>Actions</th>
Inside the foreach row:
<td>
    <a asp-action="Edit" asp-route-id="@p.Id">Edit</a>

    <form asp-action="Delete" asp-route-id="@p.Id" method="post"
style="display:inline;">
        <button type="submit">Delete</button>
    </form>
</td>
```

```
11.
```

Explanation

asp-route-id passes the id to controller actions.

Delete uses a form because it is POST.