



# POST MANAGEMENT – ASP.NET CORE MVC (THỰC HÀNH)

Tài liệu này hướng dẫn từng bước xây dựng project Post Management bằng ASP.NET Core MVC. Sinh viên chỉ cần copy – paste là chạy được.



## Mục tiêu

- Hiểu kiến trúc **MVC**
- Thực hiện **CRUD (Create – Read – Update – Delete)**
- Biết dùng **Service + Dependency Injection**
- Hiểu và tránh lỗi **HTTP 400 (Anti-Forgery Token)**

## Tạo Project

```
dotnet new mvc -n PostManagementApp
cd PostManagementApp
dotnet run
```

## Program.cs

File: Program.cs

```
using PostManagementApp.Services;

var builder = WebApplication.CreateBuilder(args);

// Đăng ký Service
builder.Services.AddScoped<IPostService, PostService>();
builder.Services.AddControllersWithViews();

var app = builder.Build();

if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}
```


```
app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseRouting();
app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Post}/{action=Index}/{id?}");

app.Run();
```

---

## Model – Post

 **File:** Models/Post.cs


```
namespace PostManagementApp.Models;

public class Post
{
    public int Id { get; set; }
    public string Title { get; set; } = string.Empty;
    public string Content { get; set; } = string.Empty;
    public DateTime CreatedAt { get; set; }
    public DateTime? UpdatedAt { get; set; }
}
```

---

## Service Layer

### 4.1 IPostService

 **File:** Services/IPostService.cs

```
using PostManagementApp.Models;

namespace PostManagementApp.Services;

public interface IPostService
{
    List<Post> GetAll();
    Post? GetById(int id);
}
```

```
void Add(Post post);  
void Update(int id, Post post);  
void Delete(int id);  
}
```

## 4.2 PostService

 File: Services/PostService.cs

```
using PostManagementApp.Models;  
  
namespace PostManagementApp.Services;  
  
public class PostService : IPostService  
{  
    private static readonly List<Post> _posts = new();  
  
    public List<Post> GetAll() => _posts;  
  
    public Post? GetById(int id)  
        => _posts.FirstOrDefault(p => p.Id == id);  
  
    public void Add(Post post)  
    {  
        post.Id = _posts.Count == 0 ? 1 : _posts.Max(p => p.Id) + 1;  
        post.CreatedAt = DateTime.Now;  
        _posts.Add(post);  
    }  
  
    public void Update(int id, Post updated)  
    {  
        var post = GetById(id);  
        if (post == null) return;  
  
        post.Title = updated.Title;  
        post.Content = updated.Content;  
        post.UpdatedAt = DateTime.Now;  
    }  
  
    public void Delete(int id)  
    {  
        var post = GetById(id);  
        if (post != null)  
            _posts.Remove(post);  
    }  
}
```

```
}  
}
```

## Controller – PostController

 File: Controllers/PostController.cs

```
using Microsoft.AspNetCore.Mvc;  
using PostManagementApp.Models;  
using PostManagementApp.Services;  
  
namespace PostManagementApp.Controllers;  
  
public class PostController : Controller  
{  
    private readonly IPostService _postService;  
  
    public PostController(IPostService postService)  
    {  
        _postService = postService;  
    }  
  
    public IActionResult Index()  
        => View(_postService.GetAll());  
  
    public IActionResult Details(int id)  
    {  
        var post = _postService.GetById(id);  
        if (post == null) return NotFound();  
        return View(post);  
    }  
  
    public IActionResult Create() => View();  
  
    [HttpPost]  
    [ValidateAntiForgeryToken]  
    public IActionResult Create(Post post)  
    {  
        if (!ModelState.IsValid) return View(post);  
        _postService.Add(post);  
        return RedirectToAction(nameof(Index));  
    }  
  
    public IActionResult Edit(int id)
```

```

{
    var post = _postService.GetById(id);
    if (post == null) return NotFound();
    return View(post);
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Edit(int id, Post post)
{
    if (id != post.Id) return NotFound();
    _postService.Update(id, post);
    return RedirectToAction(nameof(Index));
}

public IActionResult Delete(int id)
{
    _postService.Delete(id);
    return RedirectToAction(nameof(Index));
}
}

```

## Views

 Thư mục: Views/Post/

### 6.1 Index.cshtml

```

@model List<PostManagementApp.Models.Post>

<h1>Danh sách bài đăng</h1>
<a asp-action="Create" class="btn btn-primary">Tạo mới</a>

<table class="table mt-3">
<tr>
    <th>ID</th><th>Tiêu đề</th><th>Hành động</th>
</tr>
@foreach (var post in Model)
{
<tr>
    <td>@post.Id</td>
    <td>@post.Title</td>
    <td>

```

```

        <a asp-action="Details" asp-route-id="@post.Id">Chi tiết</a> |
        <a asp-action="Edit" asp-route-id="@post.Id">Sửa</a> |
        <a asp-action="Delete" asp-route-id="@post.Id"
            onclick="return confirm('Xóa?')">Xóa</a>
    </td>
</tr>
}
</table>

```

## 6.2 Create.cshtml

```

@model PostManagementApp.Models.Post

<h1>Tạo bài đăng</h1>

<form asp-action="Create" method="post">
    @Html.AntiForgeryToken()

    <div>
        <label>Tiêu đề</label>
        <input asp-for="Title" class="form-control" />
    </div>

    <div>
        <label>Nội dung</label>
        <textarea asp-for="Content" class="form-control"></textarea>
    </div>

    <button type="submit">Lưu</button>
</form>

```

## 6.3 Edit.cshtml

```

@model PostManagementApp.Models.Post

<h1>Sửa bài đăng</h1>

<form asp-action="Edit" asp-route-id="@Model.Id" method="post">
    @Html.AntiForgeryToken()
    <input type="hidden" asp-for="Id" />

    <div>

```

```
<label>Tiêu đề</label>
<input asp-for="Title" class="form-control" />
</div>

<div>
  <label>Nội dung</label>
  <textarea asp-for="Content" class="form-control"></textarea>
</div>

<button type="submit">Cập nhật</button>
</form>
```

---

## 6.4 Details.cshtml


```
@model PostManagementApp.Models.Post


<h1>@Model.Title</h1>
<p><b>Nội dung:</b></p>
<p>@Model.Content</p>
<p><i>Ngày tạo: @Model.CreatedAt</i></p>

<a asp-action="Index">Quay lại</a>
```

---

## Hoàn thành

 Sinh viên đã xây dựng xong **Post Management MVC**

 Bài tiếp theo gợi ý: - Entity Framework Core - Web API CRUD - Authentication & Authorization