Part B - Screenshots (Code + Output)

HelloWorldWebApp_Screenshot.png

HelloWorldWebApp (Part B)

Code

```
using Microsoft.AspNetCore.Mvc;

namespace HelloWorldWebApp.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index()
        {
            ViewBag.Message = "Hello, World! Welcome to ASP.NET Core MVC.";
            return View();
        }
    }
}

@{
    ViewData["Title"] = "Home";
}
<h2>@ViewBag.Message</h2>
```

Output

Browser shows: Hello, World! Welcome to ASP.NET Core MVC.

BloggingApp_Screenshot.png

BloggingApp (Part B)

Code

```
using Microsoft.AspNetCore.Mvc;
using BloggingApp.Models;
using System.Collections.Generic;

namespace BloggingApp.Controllers
{
    public class BlogController : Controller
    {
        public IActionResult Index()
        {
            var blogs = new List<Blog>
            {
                new Blog { Id = 1, Title = "First Blog", Content = "This is
my first blog post." },
                new Blog { Id = 2, Title = "Second Blog", Content = "This
is another blog post." }
            };
            return View(blogs);
        }
    }
}

@model IEnumerable<BloggingApp.Models.Blog>
<h2>Blog Posts</h2>
<ul>
@foreach (var blog in Model)
{
    <li><b>@blog.Title</b> - @blog.Content</li>
}
</ul>
```

Output

Browser shows a list of Blog Posts with two items: First Blog and Second Blog.

BookStoreAPI_Screenshot.png

BookStoreAPI (Part B)

Code

```
using Microsoft.AspNetCore.Mvc;
using BookStoreAPI.Models;
using System.Collections.Generic;

namespace BookStoreAPI.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class BooksController : ControllerBase
    {
        [HttpGet]
        public IEnumerable<Book> GetBooks()
        {
            return new List<Book>
            {
                new Book { Id = 1, Title = "C# Programming", Author =
"Author A" },
                new Book { Id = 2, Title = "ASP.NET Core", Author = "Author
B" }
            };
        }
    }
}
```

Output

```
GET /api/books -> [{"id":1,"title":"C# Programming","author":"Author
A"},{"id":2,"title":"ASP.NET Core","author":"Author B"}]
```

LoggingWithDI_Screenshot.png

LoggingWithDI (Part B)

Code

```
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Logging;

class Program
{
    static void Main(string[] args)
    {
        var serviceProvider = new ServiceCollection()
            .AddLogging(config => config.AddConsole())
            .BuildServiceProvider();

        var logger = serviceProvider.GetService<ILogger<Program>>();
        logger.LogInformation("Application started successfully!");
        logger.LogWarning("This is a warning message.");
        logger.LogError("This is an error message.");
    }
}
```

Output

```
Console output:
info: Program[0] Application started successfully!
warn: Program[0] This is a warning message.
fail: Program[0] This is an error message.
```

EFCoreSQLServer_Screenshot.png

EFCoreSQLServer (Part B)

Code

```
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.DependencyInjection;
using EFCoreSQLServer.Data;
using EFCoreSQLServer.Models;

class Program
{
    static void Main(string[] args)
    {
        var serviceProvider = new ServiceCollection()
            .AddDbContext<AppDbContext>(options =>
                options.UseSqlServer("Server=localhost;Database=EFCoreDB;Tr
usted_Connection=True;"))
            .BuildServiceProvider();

        using (var context = serviceProvider.GetService<AppDbContext>())
        {
            context.Database.EnsureCreated();
            context.Products.Add(new Product { Name = "Laptop", Price =
75000 });
            context.SaveChanges();

            foreach (var p in context.Products)
            {
                System.Console.WriteLine($"Id: {p.Id}, Name: {p.Name},
Price: {p.Price}");
            }
        }
    }
}
```

Output

```
Console shows:
Id: 1, Name: Laptop, Price: 75000
```