LAB 1
DEVELOP ASP.NET CORE WEB APP (P1)

❖ CONTENT

- Create an ASP.NET Core MVC project
- Setup database connection & implement Authentication (login/logout, register)
- Create tables with Model
- Implement CRUD features with Scaffolding technique
- Customize web layout (template) & identity pages
- Set default role for user registration
- Seed (populate) initial data to database
- Setup Authorization (role-based access) for different features

❖ INSTRUCTION

1. Open Visual Studio and create new ASP.NET Core MVC web app

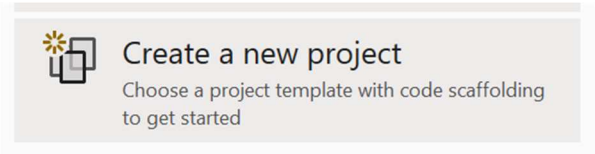

Figure 1 - Create new project in Visual Studio

2. Config ASP.NET Core project

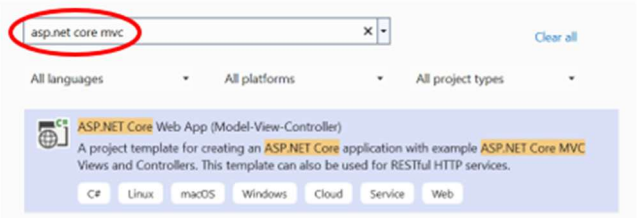

Figure 2 - Search & select ASP.NET Core Web App (MVC)

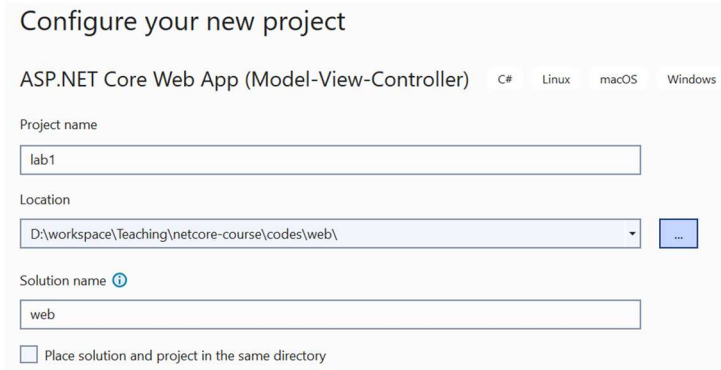

Figure 3 – Set project location, solution & project name

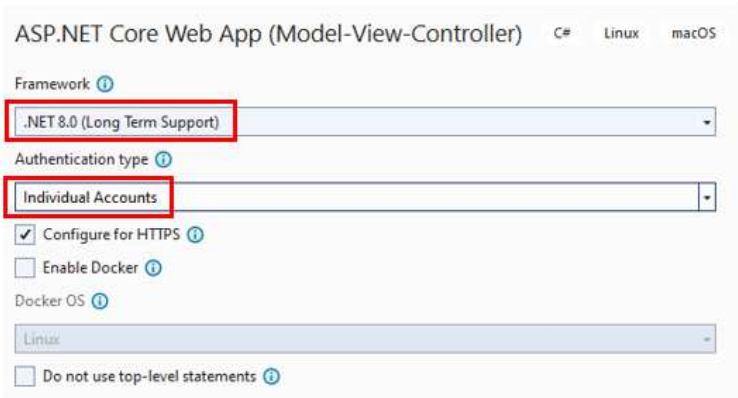

Figure 4 – Setup automatic database connection & Authentication

3. Add model to automatically create table

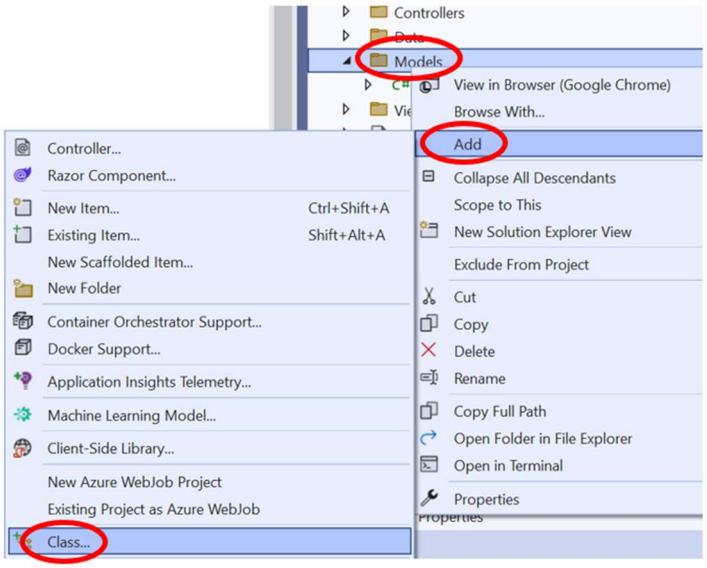

Figure 5 - Add new Model (class)

```
public class Laptop
{
    public int Id { get; set; }
    public string Brand { get; set; }
    public string Model { get; set; }
    public int Quantity { get; set; }
    public decimal Price { get; set; }
    public string Image { get; set; }
}
```

*Figure 6 – **Models/Laptop.cs***

4. Generate Controller with Views for CRUD features with Scaffolding technique

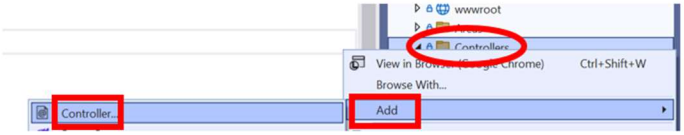

Figure 7 - Add new controller (class)

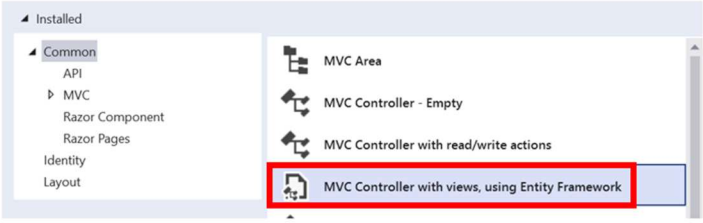

Figure 8 - Select MVC Controller with views

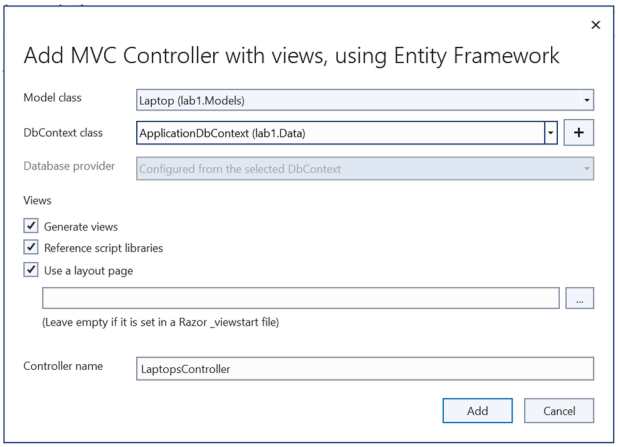

Figure 9 - Select Model class & DbContext class

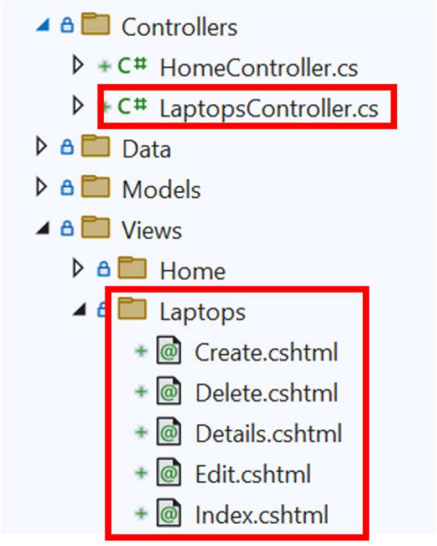

Figure 10 - Controller & Views have been generated for selected Model

5. Override Identity pages to customize view of Authentication (Login, Register,…)

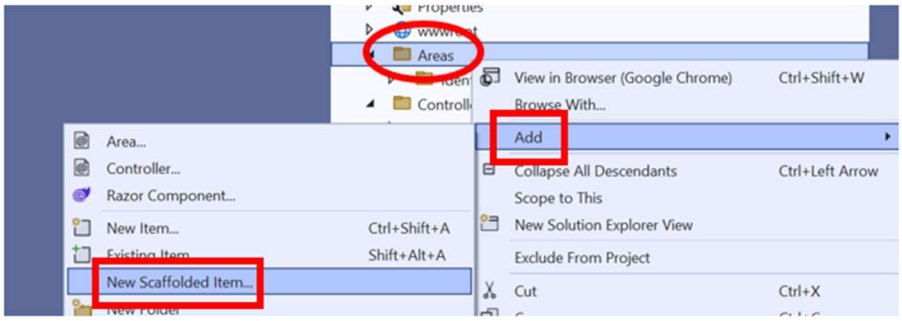

Figure 11 - Add new Scaffolded Item

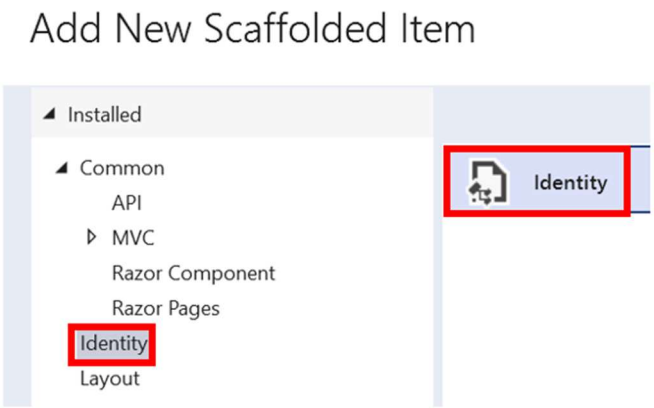

*Figure 12 - Select option **Identity***

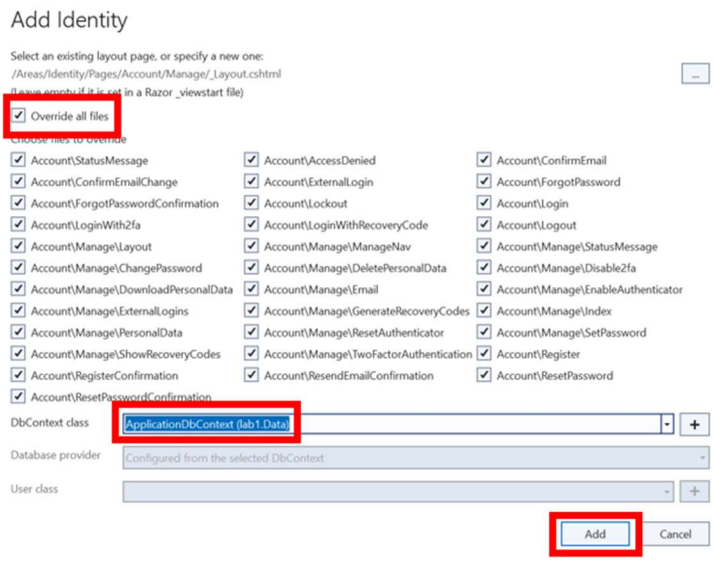

Figure 13 - Override all files & select DBContext class

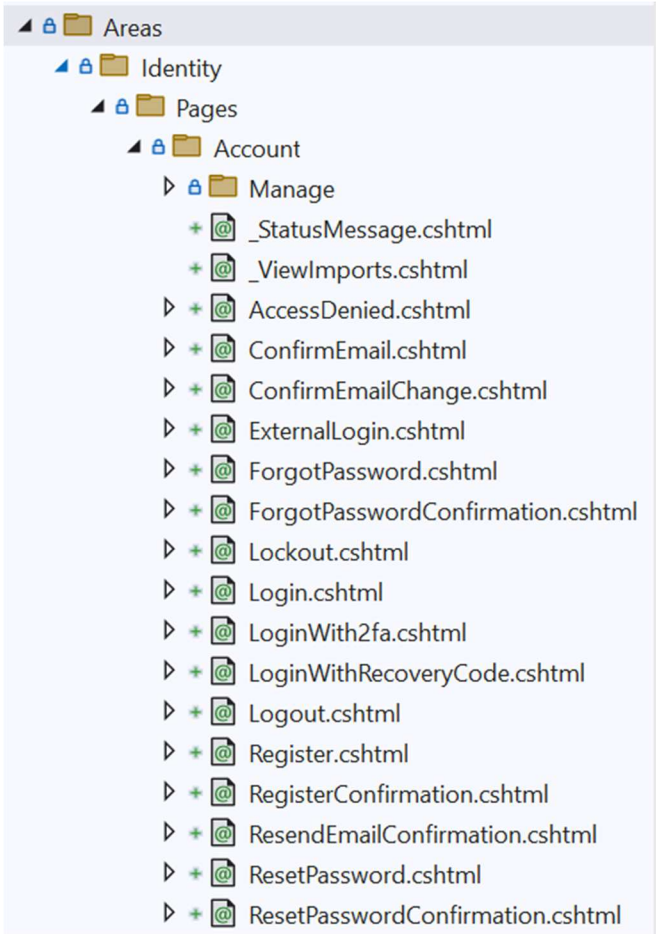

Figure 14 - All Identity pages were displayed to be overridden

6. Set default role for user registration (optional)

```
var callbackUrl = Url.Page(
    "/Account/ConfirmEmail",
    pageHandler: null,
    values: new { area = "Identity", userId = userId, code = code, returnUrl
     = returnUrl },
    protocol: Request.Scheme);

//Set default role "User" for new user registration
var role = "User";
await _userManager.AddToRoleAsync(user, role);

await _emailSender.SendEmailAsync(Input.Email, "Confirm your email",
    $"Please confirm your account by <a href='{HtmlEncoder.Default.Encode
     (callbackUrl)}'>clicking here</a>.");
```

*Figure 15 - **Areas/Identity/Pages/Account/Register.cshtml.cs***

7. Customize web layout & other web pages

```
<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
    <ul class="navbar-nav flex-grow-1">
        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-controller="Laptops" asp-action="Index">Laptop</a>
        </li>
    </ul>
    <partial name="_LoginPartial" />
</div>
```

*Figure 16 - **Views/Shared/_Layout.cshtml***

```
<div class="text-center">
    <img src="https://monamedia.co/wp-content/uploads/2023/06/dac-diem-noi-bat-cua-asp-net.jpg" />
</div>
```

*Figure 17 - **Views/Home/Index.cshtml***

```
<td>
    @Html.DisplayFor(modelItem => item.Price)
</td>
<td>
    <img src="@Html.DisplayFor(modelItem => item.Image)" width="100" height="100"/>
</td>
```

*Figure 18 - **Views/Laptops/Index.cshtml***

8. Seed initial data to database

```
protected override void OnModelCreating(ModelBuilder builder)
{
    base.OnModelCreating(builder);

    //Seed data for User & Role
    SeedUserRole(builder);

    //Seed data for table Laptop
    SeedLaptop(builder);
}
```

*Figure 19 - **Data/ApplicationDbContext.cs (1)***

```
private void SeedUserRole(ModelBuilder builder)
{
    //A) Setup IdentityUser
    //1. Create accounts
    var adminAccount = new IdentityUser
    {
        Id = "user1",
        UserName = "admin@gmail.com",
        Email = "admin@gmail.com",
        NormalizedUserName = "ADMIN@GMAIL.COM",
        NormalizedEmail = "ADMIN@GMAIL.COM",
        EmailConfirmed = true
    };

    var userAccount = new IdentityUser
    {
        Id = "user2",
        UserName = "user@gmail.com",
        Email = "user@gmail.com",
        NormalizedUserName = "USER@GMAIL.COM",
        NormalizedEmail = "USER@GMAIL.COM",
        EmailConfirmed = true
    };

    //2. Declare hasher for password encryption
    var hasher = new PasswordHasher<IdentityUser>();

    //3. Setup password for accounts
    adminAccount.PasswordHash = hasher.HashPassword(adminAccount, "123456");
    userAccount.PasswordHash = hasher.HashPassword(userAccount, "123456");

    //4. Add accounts to database
    builder.Entity<IdentityUser>().HasData(adminAccount, userAccount);
```

*Figure 20 - **Data/ApplicationDbContext.cs (2)***

```
        //B) Setup IdentityRole
        builder.Entity<IdentityRole>().HasData(
            new IdentityRole
            {
                Id = "role1",
                Name = "Administrator",
                NormalizedName = "ADMINISTRATOR"
            },
             new IdentityRole
             {
                 Id = "role2",
                 Name = "User",
                 NormalizedName = "USER"
             });

        //C) Setup IdentityUserRole
        builder.Entity<IdentityUserRole<string>>().HasData(
            new IdentityUserRole<string>
            {
                UserId = "user1",
                RoleId = "role1"
            },
            new IdentityUserRole<string>
            {
                UserId = "user2",
                RoleId = "role2"
            }
         );
    }
```

*Figure 21 - **Data/ApplicationDbContext.cs (3)***

```
private void SeedLaptop(ModelBuilder builder)
{
    builder.Entity<Laptop>().HasData(
        new Laptop
        {
            Id = 1,
            Brand = "Apple",
            Model = "Macbook Pro M2",
            Quantity = 10,
            Price = 2345,
            Image = "https://bizweb.dktcdn.net/100/444/581/products/macbook-m1-vs-intel-1536x1268-6c00654d-ad87-4caf-8b88-
              aa6c34048199.png?v=1656134590567"
        },
        new Laptop
        {
            Id = 2,
            Brand = "Dell",
            Model = "XPS 15",
            Quantity = 15,
            Price = 1999,
            Image = "https://thegioiso365.vn/wp-content/uploads/2023/04/Dell-xps-9530-3.png"

        },
        new Laptop
        {
            Id = 3,
            Brand = "LG",
            Model = "Gram 17",
            Quantity = 22,
            Price = 2024,
            Image = "https://product.hstatic.net/1000333506/product/pc-gram-17z90q-b-
              gallery-02_dd780c6249ec430b84f82ed466fffd6e.jpg"
        }
    );
}
```

*Figure 22 - **Data/ApplicationDbContext.cs (4)***

```
builder.Services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
    .AddRoles<IdentityRole>()
    .AddEntityFrameworkStores<ApplicationDbContext>();
builder.Services.AddControllersWithViews();
```

*Figure 23 - **Program.cs***

9. Setup role-based access for different features
   - Restrict user access in backend (controller)

```
//Setup role for the whole controller
[Authorize(Roles = "Administrator")]
public class LaptopsController : Controller
```

*Figure 24 - **Controllers/LaptopsController.cs (1)***

```
//Setup multiple roles for an action
[Authorize(Roles = "Administrator, User")]
public async Task<IActionResult> Index()
```

*Figure 25 - **Controllers/LaptopsController.cs (2)***

```
//Setup single role for an action
[Authorize(Roles = "Administrator")]
public IActionResult Create()
```

*Figure 26 - **Controllers/LaptopsController.cs (3)***

   - Restrict user access in frontend (view)

```
<p>
    @if (User.Identity.IsAuthenticated && User.IsInRole("Administrator"))
    {
        <a asp-action="Create">Create New</a>
    }
</p>
```

*Figure 27 - **Views/Laptops/Index.cshtml***

10.Migrate data to database

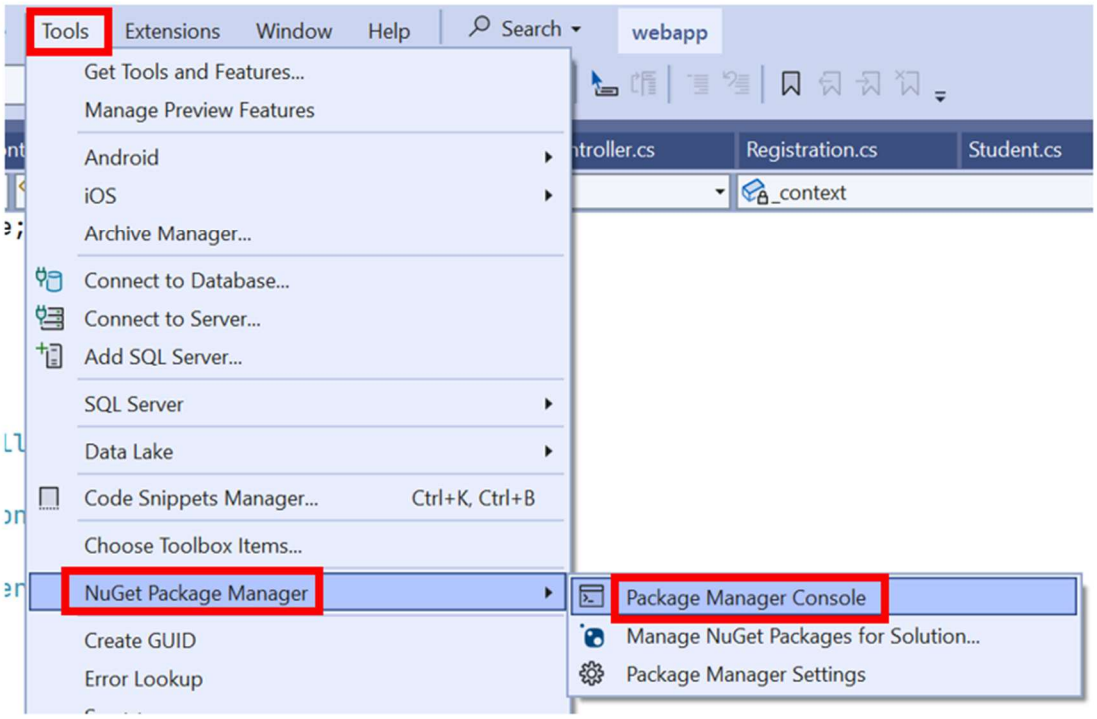

*Figure 28 - Open **Package Manager Console (PMC)***

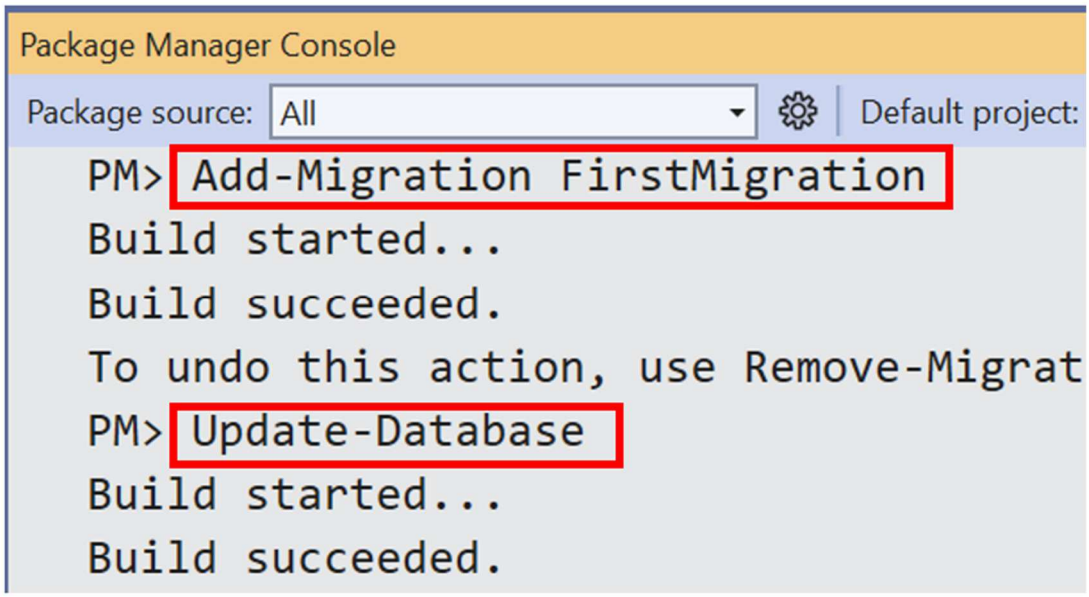

Figure 29 - Add migration & update database using PMC

11.Sample final result

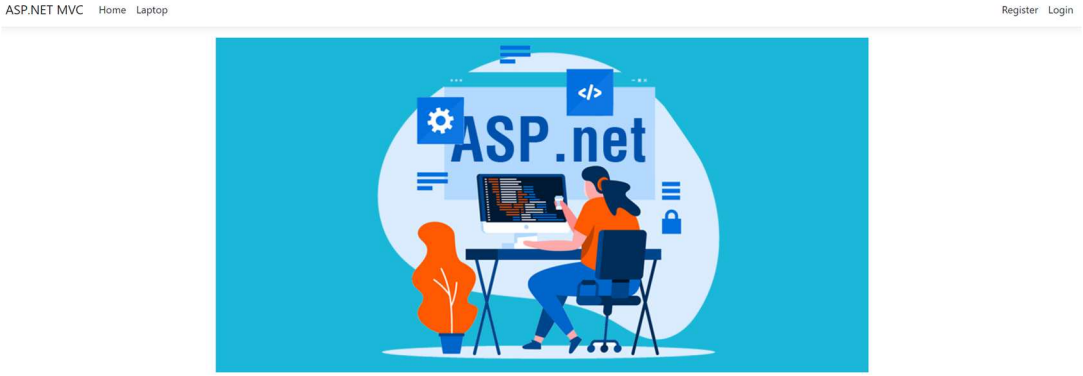

Figure 30 - Homepage

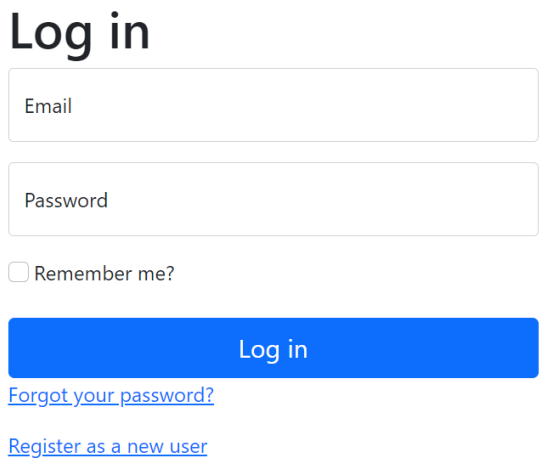

Figure 31 - Login page

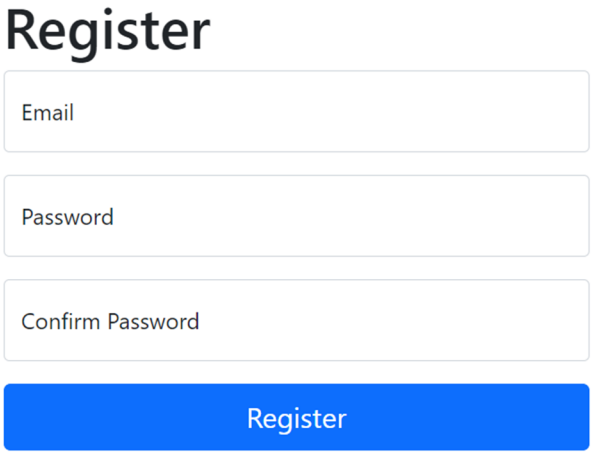

Figure 32 - Register page

Index

Create New

Brand	Model	Quantity	Price	Image	
Apple	Macbook Pro M2	10	2345.00		Edit | Details | Delete
Dell	XPS 15	15	1999.00		Edit | Details | Delete
LG	Gram 17	22	2024.00		Edit | Details | Delete

Figure 33 - View laptop list

Create

Laptop

Brand

Model

Quantity

Price

Image

Create

Back to List

Figure 34 - Create new laptop

Edit

Laptop

Brand

Apple

Model

Macbook Pro M2

Quantity

10

Price

2345.00

Image

https://bizweb.dktcdn.net/100/444/581/products/macb

Save

Back to List

Figure 35 - Edit existing laptop

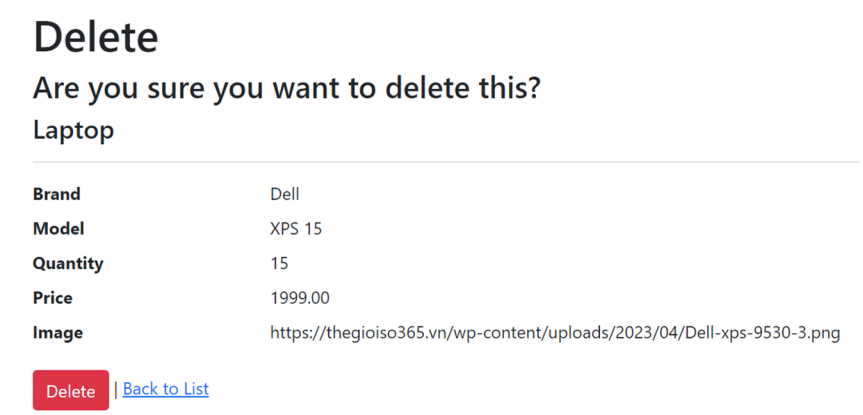

Figure 36 - Delete existing laptop

Figure 37 - Access denied page