

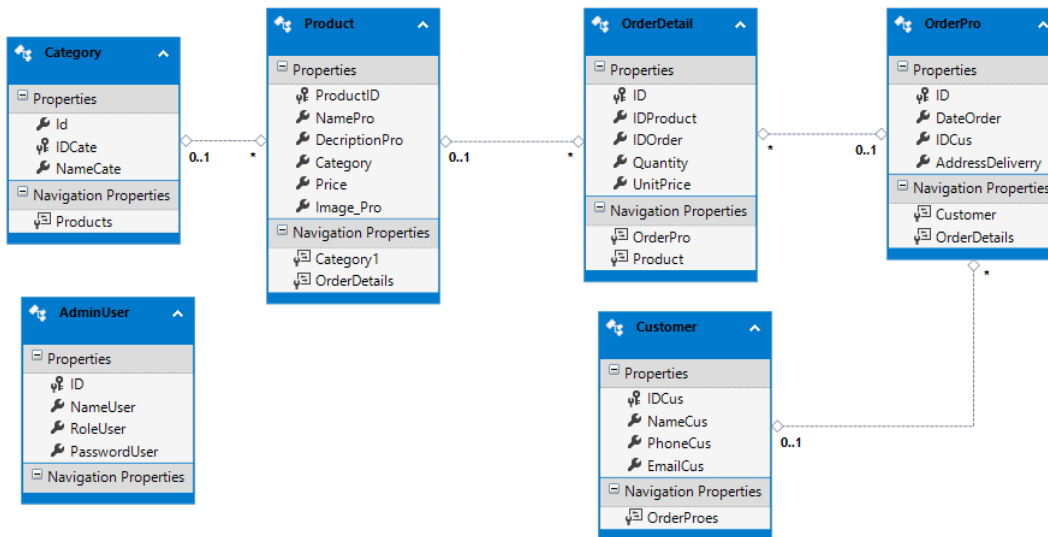
THỰC HÀNH LẬP TRÌNH WEB NÂNG CAO

BÀI THỰC HÀNH 06: Truy vấn dữ liệu với LINQ

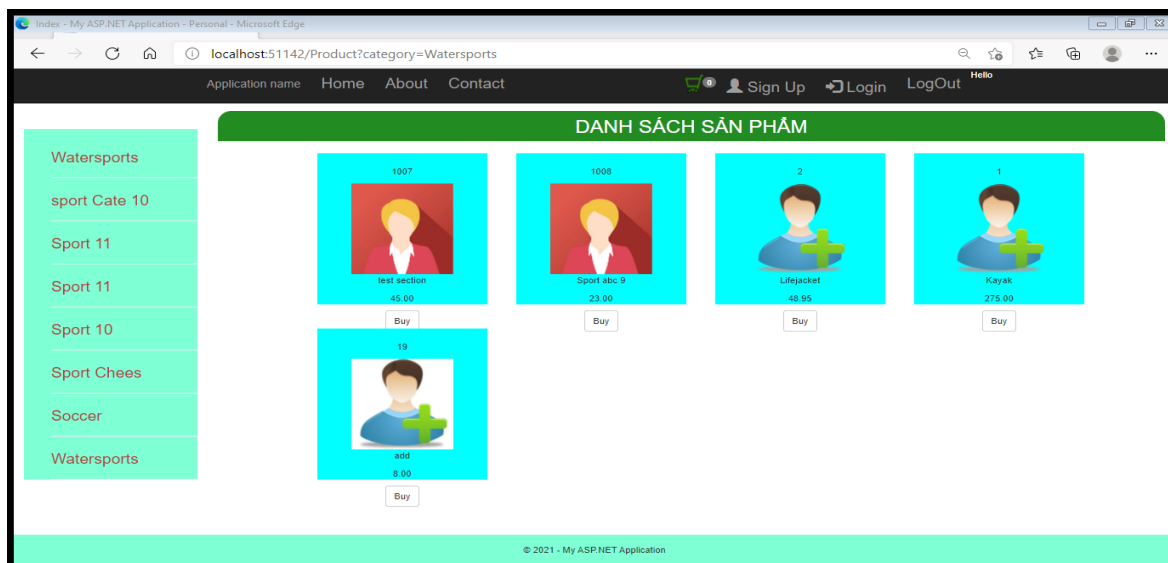
Kết nối CSDL bằng Database First

- Cơ sở dữ liệu được sử dụng trong bài thực hành: **DBSportStore**

[Link tải database](#)



Phần 1: Xây dựng Layout gắn Danh sách Category vào Layout



Bước 1: Mở file _Layout.cshtml, sử dụng Grid System chia <body> thành 2 cột 2-10

```
<div class="container-fluid body-content" >
  <div class="row">
    <div class="col-md-2">
      @{ Html.RenderAction("CategoryPartial", "Categories"); }
    </div>
    <div class="col-md-10" id="div-content">
      @RenderBody()
    </div>
  </div>
</div>
<div id="div-footer" >
  <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
</div>
```

Định dạng CSS cho #div-footer

```
<style>

  #div-footer {
    position: fixed;
    text-align: center;
    bottom: 0;
    left: 0;

    padding-top: 15px;
    height: 50px;
    width: 100%;
    background-color: aquamarine;
  }

</style>
```

Bước 2: Mở **ProductController.cs** cập nhật xử lý trong action Index()

-Nếu User chưa chọn Category Name sẽ hiển thị tất cả danh sách sản phẩm, ngược lại thì hiển thị danh sách sản phẩm theo Category.

```
public ActionResult Index(string category)
{
    if (category == null)
    {
        var productList = database.Products.OrderByDescending(x => x.NamePro);

        return View(productList);
    }
    else
    {
        var productList = database.Products.OrderByDescending(x => x.NamePro)
            .Where(x => x.Category == category);
        return View(productList);
    }
}
```

Bước 3: Viết action kiểu PartialViewResult trả về danh sách Category để gắn vào _Layout.cshtml. Viết trong **CategoryController.cs**

```
DBSportStoreEntities database = new DBSportStoreEntities();
0 references
public PartialViewResult CategoryPartial()
{
    var cateList = database.Categories.ToList();
    return PartialView(cateList);
}
```

Bước 4: Add CategoryPartial() đến View => Chọn **Create as a Partial View**

Add View

View name:
CategoryPartial

Template:
Empty

Model class:
Category (DemoContextDB.Models)

Data context class:

Options:

☒ Create as a partial view

☐ Reference script libraries

☒ Use a layout page:

~/Views/Shared/_Layout.cshtml

(Leave empty if it is set in a Razor _viewstart file)

Add
Cancel

Bước 5: Trong **CategoryPartial.cshtml** viết xử lý hiển thị NameCate và liên kết với sản phẩm thuộc Cate tương ứng

```

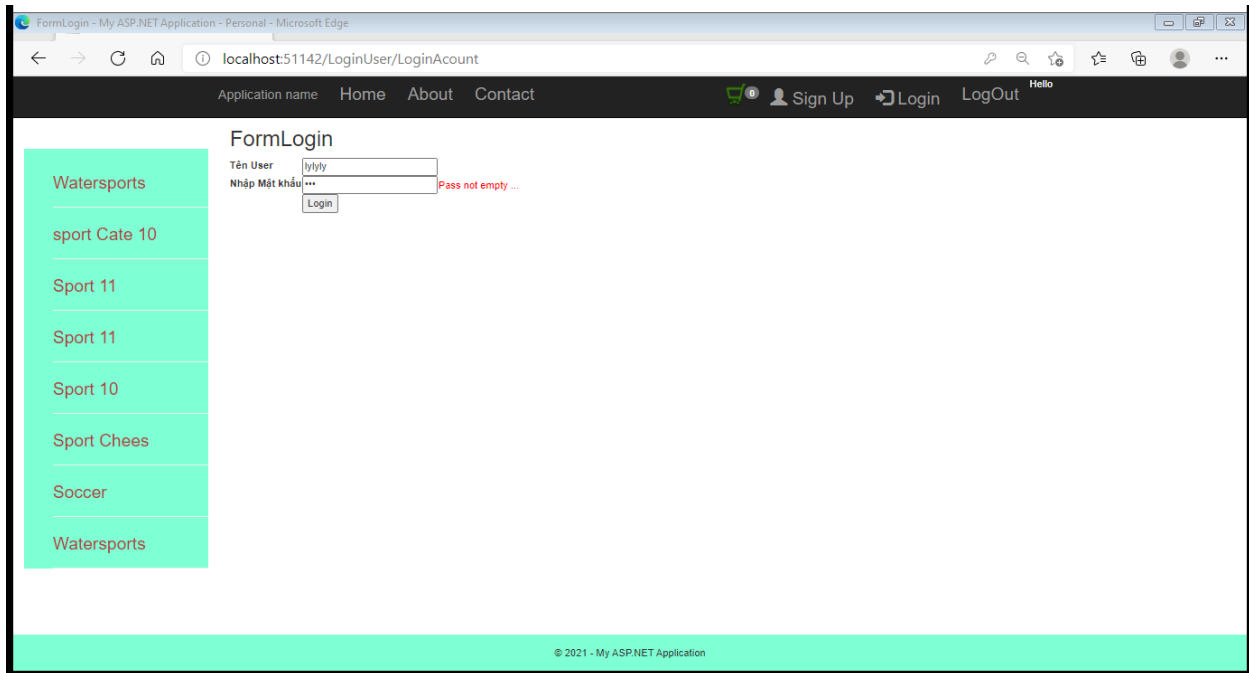
1  @using DemoContextDB.Models
2  @model List<Category>
3  <style>
4      #div-u {
5          padding-top: 30px;
6          margin-top: 50px;
7          background-color: aquamarine;
8          text-decoration: none;
9      }
10     .div-cate{
11         height: 30px;
12         width: 100%;
13     }
14     a {
15         color: #a94442;
16         text-decoration: none;
17         font-size: x-large;
18     }
19     a:hover {
20         text-decoration: none;
21     }
22 </style>
23 <ul style="list-style-type: none" id="div-u">
24     @foreach (var item in Model)
25     {
26         <li class="div-cate">
27             @Html.ActionLink(item.NameCate, "Index", "Product", new { category = item.IDCate.Trim() }, "")
28         </li>
29         <hr />
30     }
31 </ul>

```

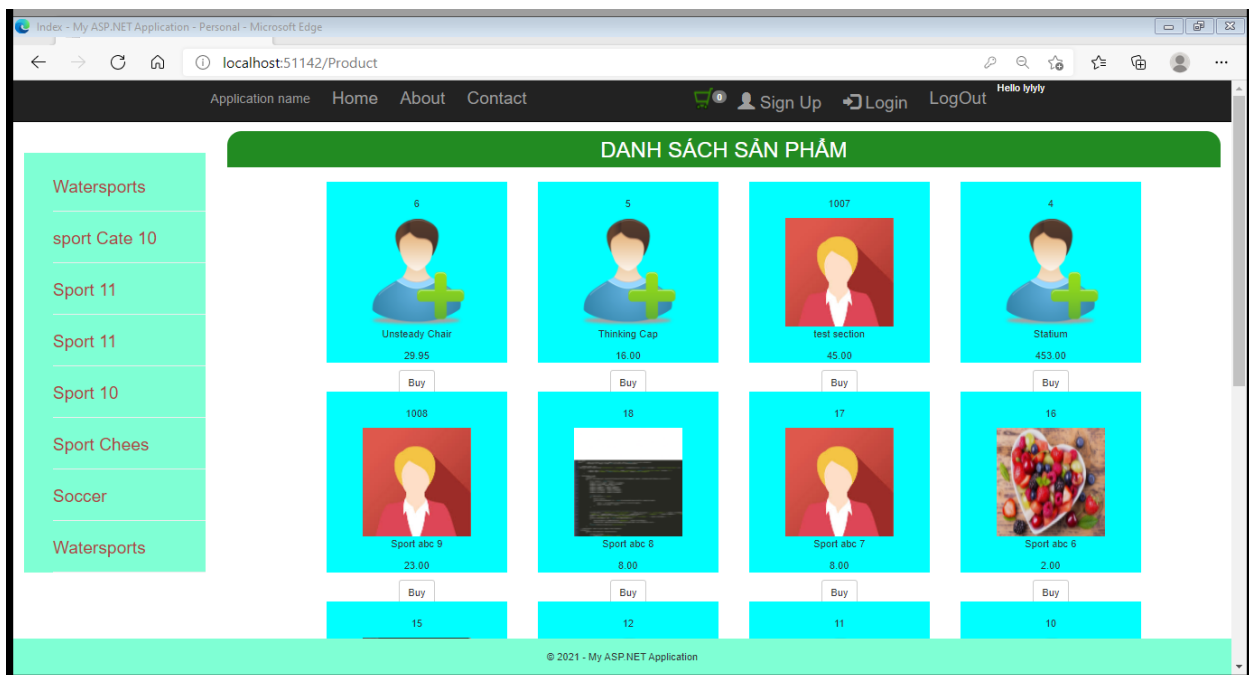
Phần 2: Hiển thị UserName lên thanh top Menu

Tình huống:

-Nếu User chưa đăng nhập thì hiển thị “Hello”



-Nếu User đăng nhập thành công thì hiển thị “Hello” +UserName



Bước 1: Review lại action `LoginAccount()` trong `LoginUserController.cs` đã làm ở bài thực hành số 03. Có 1 chỗ thay đổi là bài thực hành số 03 Login bằng ID và Password. Trong bài thực hành số 06 thì sửa lại Login bằng Username và Password

```
[HttpPost]

0 references
public ActionResult LoginAccount(AdminUser _user)
{
    var check = database.AdminUsers.
        Where(s => s.NameUser == _user.NameUser && s.PasswordUser == _user.PasswordUser).FirstOrDefault();

    if (check == null) //login sai thông tin
    {
        ViewBag.ErrorInfo = "SaiInfo";
        return View("Index");
    }
    else
    {
        database.Configuration.ValidateOnSaveEnabled = false;
        Session["NameUser"] = _user.NameUser;
        return RedirectToAction("Index", "Product");
    }
}
```

Review lại `Index()` hiển thị form Login

```
1  @model DemoContextDB.Models.AdminUser
2
3  @{
4      ViewBag.Title = "FormLogin";
5      Layout = "~/Views/Shared/_Layout.cshtml";
6  }
7  <style>
8      .field-validation-error {
9          color: red;
10     }
11 </style>
12 <h2>FormLogin</h2>
13 @using (Html.BeginForm("LoginAccount", "LoginUser"))
14 {
15     @Html.AntiForgeryToken()
16     <table>
17     <tr>
18         <td>@Html.LabelFor(model => model.NameUser)</td>
19         <td>@Html.TextBoxFor(model => model.NameUser, new { style = "field-validation-error" })</td>
20         <td>@Html.ValidationMessageFor(model => model.NameUser)</td>
21     </tr>
22     <tr>
23         <td>@Html.LabelFor(model => model.PasswordUser)</td>
24         <td>@Html.EditorFor(model => model.PasswordUser)</td>
25         <td>@Html.ValidationMessageFor(model => model.PasswordUser)</td>
26     </tr>
27     <tr>
28         <td colspan="3">
29             @if (ViewBag.ErrorInfo == null)
30             {
31                 <p>@ViewBag.ErrorInfo</p>
32             }
33         </td>
34     </tr>
35     <tr>
36         <td colspan="3">
37             <input type="submit" value="Login" />
38         </td>
39     </tr>
40 </table>
41 }
```

Bước 2: Mở _Layout.cshtml thực hiện thêm phần hiển thị UserName lên top Menu



```
<ul class="nav navbar-nav navbar-right">
  <li>
    <span>
      <a href="/ShoppingCart/ShowCart">
        
      </a>
    </span>
    @Html.Action("BagCart", "ShoppingCart")
  </li>
  <li><a href="/LoginUser/RegisterUser"><span class="glyphicon glyphicon-user"></span> Sign Up</a></li>
  <li><a href="/LoginUser/Index"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
  <li><a href="@Url.Action("LogOutUser", "LoginUser")">LogOut</a></li>
  <li class="format-hello">
    @if (Session["NameUser"] == null)
    {
      <span>Hello</span>
    }
    else
    {
      <span> <text> Hello </text> @Session["NameUser"] </span>
    }
  </li>
</ul>
```

Định dạng css của lớp .format-hello

```
li.format-hello {
  padding-top: 15px;
  color: white;
}
```