

CodeCoolShop

Krzysztof Fieber; Michał Nowakowski; Tomasz Straszewski

Cel projektu

Celem projektu było stworzenie aplikacji internetowej umożliwiającej:

- przeglądanie produktów,
- sortowanie wg dostępnych kategorii,
- dodawanie produktów do koszyka,
- dokonywanie płatności za zamówienie.

Do realizacji projektu wykorzystaliśmy:



Wyzwania

- ASP.NET FORMS
- Komunikacja .cshtml z .cs
- Przypomnienie JavaScript
- Dao

Co się nauczyliśmy?

1. Przekazywanie danych z C# do HTML poprzez:
2. Zastosowanie w praktyce wzorca DAO.
3. Serializacja i Deserializacja danych.
4. Zastosowanie metod GET i POST w ASP.NET.
5. zastosowanie Ajax w kierunku C#
6. podłączenie bazy danych
7. Użycie frameworka Entity

```
43 <script>
44     document.getElementById("@item.product.Id").addEventListener("change",
45     (event) => {
46         var quantity = document.getElementById("@item.product.Id").value;
47         var prize = document.getElementsByName("@item.product.Id");
48         console.log(quantity);
49         console.log(prize[0].innerText);
50         //var quantity = querySelector("#i").val();
51         //var prize = querySelector("#prize").val();
52         var intquant = parseInt(quantity);
53         var intprize = parseInt(prize[0].innerText);
54         var total = intprize * intquant;
55         //var child = document.createElement('div');
56         var stringtotal = total.toString();
57         //child.innerHTML = stringtotal;
58         var subtotal = document.getElementById("@item.product.Name").innerText = stringtotal;
59     })
60 </script>
```

Powtórka z Javascript w celu uzyskania obliczeń bez przeładowywania strony

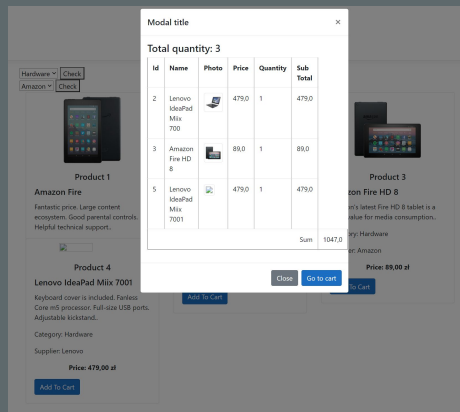
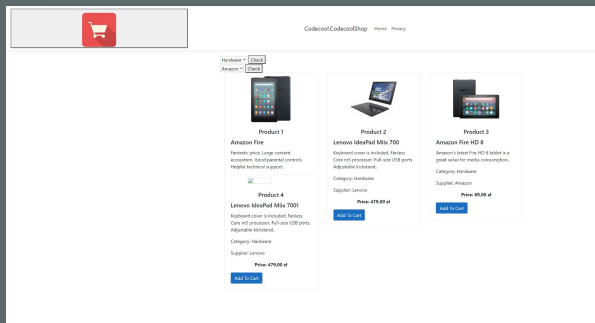
```
135 public IActionResult Quantity()  
136 {  
137     var cart = SessionHelper.GetObjectFromJson<List<LineItem>>(HttpContext.Session, key: "cart");  
138     var stringValues:ICollection = Request.Form;  
139     var count1:int = cart.Count;  
140     var count2:int = stringValues["q"].Count;  
141     List<LineItem> ClearedCart = new List<LineItem>();  
142     for (int i = 0; i < count1; i++)  
143     {  
144         var intValue = Int16.Parse(stringValues["q"][i]);  
145         cart[i].Quantity = intValue;  
146         if (cart[i].Quantity != 0)  
147         {  
148             ClearedCart.Add(cart[i]);  
149         }  
150     }  
151     SessionHelper.SetObjectAsJson(HttpContext.Session, key: "cart", ClearedCart);  
152     //ViewBag.q = cart;  
153     return View("Index1", ClearedCart);  
154 }  
155  
156
```

Przykładowa metoda IActionResult.

Entity Framework

```
29  Odwołania: 3  
30  public void Remove(int id)  
31  {  
32      context.Supplier.Remove(entity: this.Get(id));  
33  }  
  
34  Odwołania: 11  
35  public Supplier Get(int id)  
36  {  
37      return context.Supplier.Find(id);  
38  }  
  
39  Odwołania: 5  
40  public IEnumerable<Supplier> GetAll()  
41  {  
42      return context.Supplier.ToList();  
43  }  
44  }
```


Kilka Screenów



Cart Page

Total quantity: 3

Id	Name	Photo	Price	Quantity	Sub Total
2	Lenovo IdeaPad Miix 700		479.0	Quantity: <input type="text" value="5"/>	2395
3	Amazon Fire HD 8		89.0	Quantity: <input type="text" value="2"/>	178
5	Lenovo IdeaPad Miix 700i		479.0	Quantity: <input type="text" value="1"/>	479.0
Sum					1047.0

[Checkout](#)

[Continue Shopping](#)

Prezentacja strony.

Czy są jakieś
pytania?

Dziękujemy!