

ASP .NET

Visual Studio 2022

RETO2 - ASP .NET – SASKIA

WineShop web aplikazioa

Karritoa kodifikatzen:

<https://shakeelosmani.wordpress.com/2015/12/27/asp-net-mvc-step-by-step-complete-e-commerce-tutorial/>

“Session” aldagaia erabiltzen dugu zein erabiltzaile konektatuta dagoen jakiteko eta horrela karrito ezberdinak ondo identifikatuta egongo dira. Hori egiteko program.cs fitxategian lerro batzuk gehitu behar ditugu

```
builder.Services.AddControllersWithViews();
```

```
builder.Services.AddDistributedMemoryCache();
```

```
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromSeconds(900);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});
```

```
//Gure zerbitzuak
```

```
.....
```

```
app.UseRouting();
```

```
app.UseAuthentication();
app.UseAuthorization();
```

```
app.UseSession();
```

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

WineShop web aplikazioa

Erabiltzaile bakoitzak bere karrittoa du, eta zein karritotan gehitu behar diren ardoak jakiteko klase berezi bat egingo dugu: **Saskia.cs (zerbitzuen karpeta jarriko dugu)**

```
public class Saskia
{
    public string Saskiald { get; set; }
    public const string CartSessionKey = "cartId";

    public static Saskia SaskiaLortu(HttpContext contestua)
    {
        var cart = new Saskia();
        cart.Saskiald = cart.SaskiaLortulId(contestua);
        return cart;
    }
    public static Saskia SaskiaLortu(Controller controller)
    {
        return SaskiaLortu(controller.HttpContext);
    }
    public string SaskiaLortulId(HttpContext contestua)
    {
        if (contestua.Session.GetString(CartSessionKey) == null)
        {
            if (!string.IsNullOrEmpty(contestua.User.Identity.Name))
            {
                contestua.Session.SetString(CartSessionKey, contestua.User.Identity.Name);
            }
            else
            {
                Guid tempCartId = Guid.NewGuid();
                contestua.Session.SetString(CartSessionKey, tempCartId.ToString());
            }
        }
        return contestua.Session.GetString(CartSessionKey);
    }
}
```

WineShop web aplikazioa

Gero Kontroladorea sortu (zerbitzua erabiliko du: saskiaService)

SaskiaController

```
using Microsoft.AspNetCore.Mvc;
using WineShop.Services;
using WineShop.ViewModels;

namespace WineShop.Controllers
{
    public class SaskiaController : Controller
    {
        private readonly ISaskiaService _saskiaService;
        public SaskiaController(ISaskiaService saskiaService)
        {
            _saskiaService = saskiaService;
        }
        public async Task<IActionResult> Index()
        {
            return View(); // momentuz bista hutsa sortuko dugu (empty txantiloia)
        }
        public async Task<IActionResult> SaskiaGehitu(int id)
        {
            var cart = Saskia.SaskiaLortu(this.HttpContext); //aurretik sortu dugun Saskia klasea erabiliz
            await _saskiaService.SaskiaGehitu(id, cart.Saskiald); //zerbitzu berrian karritoan gehitzeko
            return RedirectToAction("Index", new { id = cart.Saskiald });
        }
    }
}
```

WineShop web aplikazioa

Zerbitzu berria: SaskiaService. Lehenengo Interfacea sortuko dugu

```
public interface ISaskiaService
{
    Task SaskiaGehitu(int ardoald, string saskiald);}
```

WineShop web aplikazioa**Zerbitzu berria: SaskiaService**

```

public class SaskiaService
{
    private Uri rutaTodos = new Uri("https://localhost:44367/api/SaskiaAlea/");

    private readonly IArdoaService _ardoaService;
    public SaskiaService(IArdoaService ardoaService)
    {
        _ardoaService = ardoaService;
    }
    public async Task SaskiaGehitu(int ardoald, string saskiald)
    {
        //Saskia existitzen den begiratu
        Uri rutasaskia = new Uri(rutaTodos, saskiald);
        List<SaskiaAlea> saskiaAleaList = new List<SaskiaAlea>();
        using (var httpClient = new HttpClient())
        {
            using (var response = await httpClient.GetAsync(rutasaskia))
            {
                string apiResponse = await response.Content.ReadAsStringAsync();
                saskiaAleaList = JsonConvert.DeserializeObject<List<SaskiaAlea>>(apiResponse);
            }
        }
        SaskiaAlea cartitem= new SaskiaAlea();
        cartitem = null;
        if (saskiaAleaList != null)
        {
            cartitem = saskiaAleaList.FirstOrDefault(s => s.Ardoald == ardoald);
        }
    }
}

```

Program.cs

```
services.AddScoped<ISaskiaService, SaskiaService>();
```

WineShop web aplikazioa**Zerbitzu berria: SaskiaService**

```
if (saskiaAleaList == null || cartitem == null)    {
    // Karrito berria sortu
    SaskiaAlea cartitem = new SaskiaAlea
    {
        Ardoald = ardoald,
        Saskiald = saskiald,
        Kantitatea = 1,
        Data = DateTime.Now
    };
    using (var httpClient = new HttpClient())
    {
        StringContent content = new StringContent(JsonConvert.SerializeObject(cartitem), Encoding.UTF8,
"application/json");
        var response = await httpClient.PostAsync(rutaTodos, content);
        response.EnsureSuccessStatusCode();
    }
}
else
{
    cartitem.Kantitatea++;
    using (var httpClient = new HttpClient())
    {
        StringContent content = new StringContent(JsonConvert.SerializeObject(cartitem), Encoding.UTF8,
"application/json");
        var response = await httpClient.PutAsync(rutasaskia, content);
        response.EnsureSuccessStatusCode();
    }
}
}
```

WineShop web aplikazioa**Models berria: SaskiaAlea**

```
public class SaskiaAlea
{
    [Key]
    public int Id { get; set; }
    public string Saskiald { get; set; }
    public int Kantitatea { get; set; }
    public System.DateTime Data { get; set; }
    public int Ardoald { get; set; }
}
```


WineShop web aplikazioa

REST API-an aldaketak batzuk egin behar ditugu saskia irakurtzeko (Saskiald erabili behar da)

ISaskiaAleaService

```
public interface ISaskiaAleaService
{
    ....
    Task<List<SaskiaAlea>> GetSaskiaAlea(string id);
}
```

SaskiaAleaService

```
public async Task<List<SaskiaAlea>> GetSaskiaAlea(string id)
{
    return await _context.SaskiaAlea
        .Where(a => a.Saskiald == id).ToListAsync();
}
```

SaskiaAleaController

```
public async Task<ActionResult<List<SaskiaAlea>>> GetSaskiaAlea(string id)
{
    return await _saskiaAleaService.GetSaskiaAlea(id);
}
```

WineShop web aplikazioa

Ardo baten informazioa lortuko dugu. Lehenengo pausua eredu berri bat egitea izango da, ardoaren motaren informazio ere irakurriko dugulako. Ardoa eta Mota arteko erlazioa jarriko dugu.

Mota:

```
public class Mota
{
    [Key]
    public int Id { get; set; }
    public string Izena { get; set; }
    public virtual IList<Ardoa> Ardoak { get; set; }
}
```

Ardoa:

```
Public class Ardoa
{
    .....
    public int Motald { get; set; }
    public virtual Mota Mota { get; set; }
```

WineShop web aplikazioa

Zerbitzua prestatuko dugu informazio hori jasotzeko

IArdoaService:

Task<Ardoa> GetArdoa(int id);

ArdoaService:

```
public async Task<Ardoa> GetArdoa(int id)
{
    Ardoa ardoa = new Ardoa();
    Uri rutaArdoBat = new Uri(rutaTodos, id.ToString());
    using (var httpClient = new HttpClient())
    {
        using (var response = await httpClient.GetAsync(rutaArdoBat))
        {
            string apiResponse = await response.Content.ReadAsStringAsync();
            ardoa = JsonConvert.DeserializeObject<Ardoa>(apiResponse);
        }
    }
    return ardoa;
}
```






WineShop web aplikazioa

Home/Index.cshtml orrian esteka gehituko dugu (karruseleko ardoei)

```
<div class="separator clear-left">
  <p class="btn-details">
    <i class="fa fa-list"></i><a href="" class="d-sm-none d-md-block">Detaile gehiago</a>
  </p>
  <p class="btn-add">
    <i class="fa fa-shopping-cart"></i>
    <a asp-controller="Saskia" asp-action="SaskiaGehitu" asp-route-id="@item.Id">Karritora gehitu</a>
  </p>
</div>
```

WineShop web aplikazioa

Saskiaren datuak bistaratuko ditugu:

<div>  <div>Search </div> </div>				
Izena	Irudia	Kantitatea	Salneurria	Guztira
Remelluri zuria		2	45,00	90,00
Lindes de Remelluri		1	11,40	11,40
Veuve Clicquot Brut		1	32,55	32,55
<div> <div>Bueltatu</div> <div>Ordaindu</div> </div>			Guztira: 133,95 €	

WineShop web aplikazioa

SaskiaService, metodo berri bat. Lehenengo Interfacea aldatuko dugu

```
public interface ISaskiaService
{
    .....
    Task<List<SaskiaAlea>> SaskiaLortuAleak(string saskiald);
}
```

Eta gero zerbitzua:

```
public class SaskiaService : ISaskiaService
{
    .....
    public async Task<List<SaskiaAlea>> SaskiaLortuAleak(string saskiald)
    {
        //Saskia lortu
        Uri rutasaskia = new Uri(rutaTodos, saskiald);
        List<SaskiaAlea> saskiaAleaList = new List<SaskiaAlea>();
        using (var httpClient = new HttpClient())
        {
            using (var response = await httpClient.GetAsync(rutasaskia))
            {
                string apiResponse = await response.Content.ReadAsStringAsync();
                saskiaAleaList = JsonConvert.DeserializeObject<List<SaskiaAlea>>(apiResponse);
            }
        }
        return saskiaAleaList;
    }
}
```

WineShop web aplikazioa

ViewModel bi sortuko ditugu, behar ditugun datuekin bakarrik (ViewModels karpeta berria sortuko dugu)

SaskiaAleaViewModel

```
namespace WineShop.ViewModels
{
    public class SaskiaAleaViewModel
    {
        public int Ardoald { get; set; }
        public string Izena { get; set; }
        public string Irudia { get; set; }
        public int Kantitatea { get; set; }
        public decimal Salneurria { get; set; }
    }
}
```

SaskiaViewModel

```
namespace Wineshop.ViewModels
{
    public class SaskiaViewModel
    {
        public IList<SaskiaAleaViewModel> SaskiaAleaVMList { get; set; }
    }
}
```

WineShop web aplikazioa

Gero Kontroladorean Index metodoa aldatuko dugu, karritoan dauden erosketak bistaratzeko:

SaskiaController

```
public class SaskiaController : Controller
{
    private readonly ISaskiaService _saskiaService;
    private readonly IArdoaService _ardoaService;
    public SaskiaController(ISaskiaService saskiaService, IArdoaService ardoaService)
    {
        _saskiaService = saskiaService;
        _ardoaService = ardoaService;
    }
    public async Task<ActionResult> Index(string id)
    {
        IList<SaskiaAlea> saskiaAleaList = new List<SaskiaAlea>();
        saskiaAleaList = await _saskiaService.SaskiaLortuAleak(cart.Saskiald);

        //Ardo bakoitzaren datuak hartu eta ViewModel bezala sortu
        IList<SaskiaAleaViewModel> saskiaAleaVMList = new List<SaskiaAleaViewModel>();
        foreach (var saskiaAlea in saskiaAleaList)
        {
            var ardoa = await _ardoaService.GetArdoa(saskiaAlea.Ardoald);
            SaskiaAleaViewModel saskiaAleaViewModel = new SaskiaAleaViewModel()
            {
                Ardoald = ardoa.Id,
                Irudia = ardoa.Irudia,
                Izena = ardoa.Izena,
                Kantitatea = saskiaAlea.Kantitatea,
                Salneurria = ardoa.Salneurria
            };
            saskiaAleaVMList.Add(saskiaAleaViewModel);
        }

        var saskiaViewModel = new SaskiaViewModel(); //Bistaratu dugun ViewModel osoa
        saskiaViewModel.SaskiaAleaVMList =saskiaAleaVMList;
        saskiaViewModel.SaskiaGuztira = guztira;

        return View(saskiaViewModel);
    }
}
```


WineShop web aplikazioa

Views karpeta barruan, “Saskia” karpetan sortu genuen Index.cshtml aldatuko dugu :

```
@using System.Globalization;
@model Wineshop.ViewModels.SaskiaViewModel
```

```
@{
    ViewData["Title"] = "Index";
}
<div class="container">
    <div id="update-message"></div>
    <table class="table table-striped">
        <thead>
            <tr class="borgona">
                <th>Izena</th>
                <th>Irudia</th>
                <th>Kantitatea</th>
                <th>Salneurria</th>
                <th>Guztira</th>
            </tr>
        </thead>
        @{}
            decimal saskiaguztira = 0;
        }
```

***hurrengo diapositiban jarraitzen du

Site.css

```
.altuera {
    max-height: 100px;
}
.btn-borgona {
    background-color: #981212;
}
.btn-borgona:hover {
    background-color: #f2c5c5;
    color: #FFF;
}
```

WineShop web aplikazioa

```
@foreach (var item in Model. SaskiaAleaVMList)
{
    <tr id="row-@item.Ardoald">
        <td>
            @item.Izena
        </td>
        <td>
            
        </td>
        <td>
            @item.Kantitatea
        </td>
        <td>
            @item.Salneurria
        </td>
        <td>
            @{ decimal guztira = item.Salneurria * item.Kantitatea;
               saskiaguztira += guztira;}
            @guztira
        </td>
    </tr>
}
<tr>
    <td><a class="btn btn-borgona text-light" asp-controller="Home" asp-action="Index">Bueltatu</a>
    <a class="btn btn-borgona text-light" asp-controller="Ordaindu" asp-action="Index">Ordaindu</a></td>
    <td></td>
    <td></td>
    <td class="float-right"><p>Guztira: <span id="cart-total">@Model.SaskiaGuztira.ToString("C2", CultureInfo.CurrentCulture)</span></p></td>
</tr>
</table>
</div>
```

WineShop web aplikazioa

Ajax erabilita kantitatea aldatuko dugu.

Saskia/Index.cshtml:

```
<td>
  <div>
    <a href="" class="RemoveLink" data-id="@item.Ardoald">
      <i class="fas fa-cart-arrow-down"></i>
    </a>
    <span id="item-count-@item.Ardoald"> @item.Kantitatea</span>
    <a href="" class="AddLink" data-id="@item.Ardoald" data-kantitatea="@item.Kantitatea" data-salneurria="@item.Salneurria">
      <i class="fas fa-cart-plus"></i>
    </a>
  </div>
</td>
.....
.....
```

WineShop web aplikazioa**Site.js:**

```

/*Saskira gehitzeko*/
$(document).ready(function () {
    $(".AddLink").click(function (e) {
        e.preventDefault();
        // Get the id from the link
        var recordToAdd = $(this).attr("data-id");
        var kantitatea = parseInt($(this).attr("data-kantitatea"));
        var salneurria = parseFloat($(this).attr("data-salneurria"));
        var guztira = parseFloat($("#cart-total").text());
        if (recordToAdd !== "") {
            // Perform the ajax post
            $.post("/Saskia/SaskiaGehituAjax", { "id": recordToAdd, "kantitatea": kantitatea, "salneurria": salneurria, "guztira": guztira },
                function (data) {
                    // Successful requests get here
                    // Update the page elements
                    $("#item-count-" + recordToAdd).text(data.kantitatea.toString());
                    $("#cart-total").text((data.guztira + data.salneurria).toString());
                    $("#update-message").text(data.mezua);
                });
        }
    });
})

```

WineShop web aplikazioa

SaskiaController:

```
public async Task<IActionResult> SaskiaGehituAjax(int id, int kantitatea, float salneurria, float guztira)
{
    var cart = Saskia.SaskiaLortu(this.HttpContext);
    await _saskiaService.SaskiaGehitu(id, cart.Saskiald);
    kantitatea++;
    var results = new
    {
        mezua= "Zure saskia eguneratu da",
        kantitatea = kantitatea,
        salneurria = salneurria,
        guztira = guztira
    };

    return Json(results);
}
```

WineShop web aplikazioa

Ariketa:

Saiatu karritotik kentzeko botoia prestatzen **ajax** erabilia
(gehitzeko botoiaren antzeko zerbait egin behar da)

WineShop web aplikazioa

Kontroladore berria sortzen dugu (Controlador MVC: vacio)

Kontroladore hau bezeroen datuak jasotzeko erabiliko dugu, beraz, formulario bat erakutsiko dugu datuak sartzeko.

Authorize jartzen dugu kontroladorean erabiltzaile erregistratuak bakarrik sartzeko.

OrdainduController.cs

```
using WineShop.Data;
using WineShop.Models;
using Microsoft.AspNetCore.Authorization;
using WineShop.Services;

namespace WineShop.Controllers
{
    [Authorize]
    public class OrdainduController : Controller
    {
        private readonly ISaskiaService _saskiaService;

        public OrdainduController(ISaskiaService saskiaService)
        {
            _saskiaService = saskiaService;
        }

        public IActionResult Index()
        {
            return View();
        }
    }
}
```

WineShop web aplikazioa**Eredu berria sortu Models karpeta barruan****BezeroaEskaera.cs**

```
namespace WineShop.Models
{
    public class BezeroaEskaera
    {
        [Key]
        [ScaffoldColumn(false)]
        public int Id { get; set; }
        [Required(ErrorMessage = "Izena derrigorrezkoa da")]
        public string Izena { get; set; }
        [Required(ErrorMessage = "Abizena derrigorrezkoa da")]
        public string Abizena { get; set; }
        [Required(ErrorMessage = "Helbidea derrigorrezkoa da")]
        public string Helbidea { get; set; }
        [Required(ErrorMessage = "Hiria derrigorrezkoa da")]
        public string Hiria { get; set; }
        [Required(ErrorMessage = "Herrialdea derrigorrezkoa da")]
        public string Herrialdea { get; set; }
        [Required(ErrorMessage = "Postakodea derrigorrezkoa da")]
        public string Postakodea { get; set; }
        [Required(ErrorMessage = "Telefonoa derrigorrezkoa da")]
        public string Telefonoa { get; set; }
        [ScaffoldColumn(false)]
        public System.DateTime Data { get; set; }
        [ScaffoldColumn(false)]
        public string? Erabiltzailea { get; set; }
    }
}
```


WineShop web aplikazioa

Bista berria sortu behar da «return View» gainean jarrita saguarekin eskuineko botoiarekin. «Agregar vista de Razor» aukeratu behar da:

Izena: Index

Plantilla: Create

Clase de modelo: BezeroaEskaera

×

Agregar Vista de Razor

Nombre de vista

Index

Plantilla

Create

Clase de modelo

BezeroaEskaera (Wineshop.Models)

Clase de contexto de datos

<Obligatorio>

Opciones

☐ Crear como vista parcial

☒ Hacer referencia a bibliotecas de scripts

☒ Usar página de diseño

...

(Dejar en blanco si se define en un archivo _viewstart de Razor)

Agregar

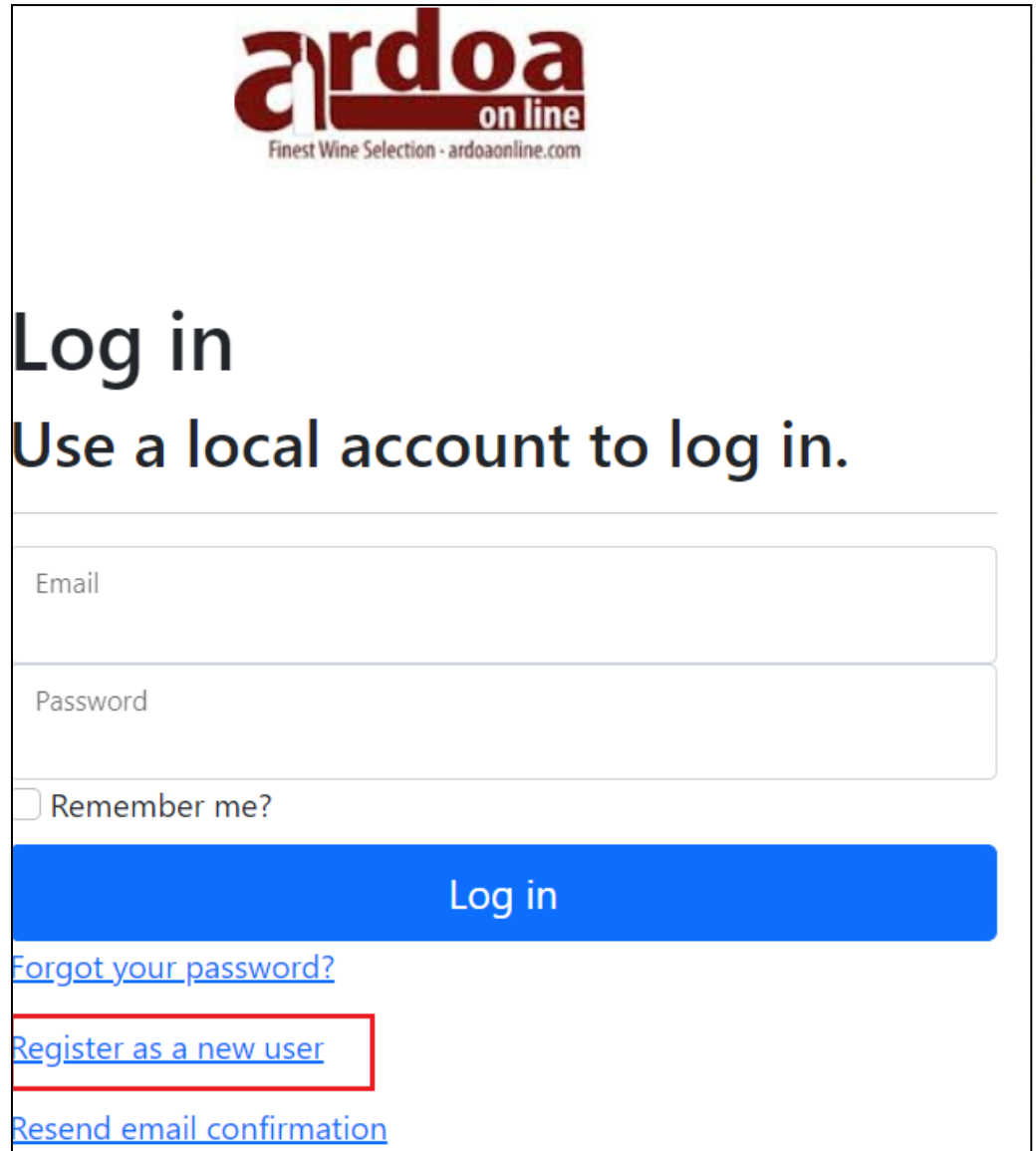
Cancelar

WineShop web aplikazioa

Ordaindu botoian klik eginda,
logeatzeko eskatuko digu, eta
konturik ez dugunez,
erregistratu beharko gara:

Datu erraz batzuk aukeratu,
adibidez, **izaskun@gmail.com**,
eta pasahitza (konplexutasun
arauak eskatzen ditu)

izaskun_1234



The screenshot shows the login interface for 'ardoa on line'. At the top is the logo with the text 'Finest Wine Selection · ardoaonline.com'. Below the logo, the text 'Log in' is displayed in a large font, followed by 'Use a local account to log in.' in a smaller font. There are two input fields: 'Email' and 'Password'. Below these fields is a checkbox labeled 'Remember me?'. A large blue button labeled 'Log in' is positioned below the checkbox. At the bottom of the form, there are three links: 'Forgot your password?', 'Register as a new user' (which is highlighted with a red rectangular box), and 'Resend email confirmation'.

WineShop web aplikazioa

Pantaila hau irtetzen badu: «Apply Migrations» botoian klikatu eta gero aplikazioa berabiarazi. Ezin bada login egin, saiatu berriro register egiten.

A database operation failed while processing the request.

SqlException: Cannot open database "aspnet-Winshop-5924a00f-3860-41a0-91e6-cb54c0bb2b5a" requested by the login. The login failed. Login failed for user 'IZASKUN-PC\izaskun'.

Applying existing migrations may resolve this issue

There are migrations that have not been applied to the following database(s):

ApplicationDbContext

- 0000000000000000_CreatelntegritySchema

Apply Migrations

In Visual Studio, you can use the **Package Manager Console** to apply pending migrations to the database:

```
PM> Update-Database
```

Alternatively, you can apply pending migrations from a command prompt at your project directory:

```
> dotnet ef database update
```

Amaieran:

Egin klik konfirmatzeko



Register confirmation

This app does not currently have a real email sender registered, see [these docs](#) for how to configure a real

WineShop web aplikazioa

Bueltatu hasierara berriro eta karritora sartu, ordaindu botoian klik egin, login egitea eskatzen du.

Ordaintzeko formularioa agertzen da

Index

Ordaintzeko formularioa

[Back to List](#)

WineShop web aplikazioa

Kontroladorea aldatuko dugu formularioa beteta bueltatzen denerako

OrdainduController.cs

namespace WineShop.Controllers

{

.....

.....

[HttpPost]

[ValidateAntiForgeryToken]

```
public async Task<IActionResult> Index([Bind("Abizena,Helbidea,Herrialdea,Hiria,Izena,Postakodea,Telefonoa")]
BezeroaEskaera bezeroaEskaera)
```

{

if (ModelState.IsValid)

{

/*Bezeroen datuak gorde*/

bezeroaEskaera.Erabilitzailea = HttpContext.User.Identity.Name;

bezeroaEskaera.Data = DateTime.Now;

await _saskiaService.EskaeraBezeraGehitu(bezeroaEskaera);

/*Eskaera gorde*/

var cart = Saskia.SaskiaLortu(this.HttpContext);

await _saskiaService.EskaeraSortu(bezeroaEskaera, cart.Saskiald);

/*Beste pantaila batera berbideratzen da*/

```
return RedirectToAction("Osatu", new { bezeroa = bezeroaEskaera.Izena + " " + bezeroaEskaera.Abizena,
saskiald = cart.Saskiald });
```

}

return View(bezeroaEskaera);

}

WineShop web aplikazioa

Eredu berri bat sortu Models karpeta barruan.

Erosketa.cs

```
public class Erosketa
{
    [Key]
    public int Id { get; set; }
    public int Kantitatea { get; set; }
    public int Ardoald { get; set; }
    public int BezeroaEskaerald { get; set; }
}
```

WineShop web aplikazioa

Zerbitzuan gehitzen ditugun metodoak

ISaskiaService.cs

```
Task EskaeraBezeroaGehitu(BezeroaEskaera bezeroaEskaera);
Task EskaeraSortu(BezeroaEskaera bezeroaEskaera, string saskiald);
```

SaskiaService.cs

```
public async Task EskaeraBezeroaGehitu(BezeroaEskaera bezeroaEskaera)
{
    Uri rutaBezeroEskaera = new Uri("https://localhost:44367/api/BezeroaEskaera/");
    using (var httpClient = new HttpClient())
    {
        StringContent content = new StringContent(JsonConvert.SerializeObject(bezeroaEskaera), Encoding.UTF8,
"application/json");
        var response = await httpClient.PostAsync(rutaBezeroEskaera, content);
        response.EnsureSuccessStatusCode();
    }
}
```

WineShop web aplikazioa**Zerbitzuan gehitzen ditugun metodoak****SaskiaService.cs**

```
public async Task EskaeraSortu(BezeroaEskaera bezeroaEskaera, string saskiald)
{
    var cartItems = await SaskiaLortuAleak(saskiald);

    foreach (var item in cartItems)
    {
        var erosketa = new Erosketa
        {
            Ardoald = item.Ardoald,
            BezeroaEskaerald = bezeroaEskaera.Id,
            Kantitatea = item.Kantitatea
        };
        Uri rutaErosketa = new Uri("https://localhost:44367/api/Erosketa/");
        using (var httpClient = new HttpClient())
        {
            StringContent content = new StringContent(JsonConvert.SerializeObject(erosketa), Encoding.UTF8,
"application/json");
            var response = await httpClient.PostAsync(rutaErosketa, content);
            response.EnsureSuccessStatusCode();
        }
    }
}
```


WineShop web aplikazioa

Behin erosketa eginda, beste pantaila batean, erosi diren ardoen informazioa bistaratuko dugu. Horretarako **ViewModel** berri bat sortuko dugu.

OsatuViewModel.cs

```
using WineShop.Models;

namespace WineShop.ViewModels
{
    public class OsatuViewModel
    {
        public string Bezeroa { get; set; }
        public string Saskiald { get; set; }
        public IList<SaskiaAlea> SaskiaAleak { get; set; }
    }
}
```

WineShop web aplikazioa

Eta OrdainduController barruan metodo berri bat gehitu behar da: Osatu

```
using Wineshop.ViewModels;
```

```
.....
```

```
public async Task<IActionResult> Osatu(string bezeroa, string saskiald)
{
    var osatuViewModel = new OsatuViewModel(); //ViewModel bat erabiliko dugu
    osatuViewModel.SaskiaAleak = await _saskiaService.SaskiaLortuAleak(saskiald);
    osatuViewModel.Saskiald = saskiald;
    osatuViewModel.Bezeroa = bezeroa;
    return View(osatuViewModel);
}
```

WineShop web aplikazioaOsatu metodoaren bista sortuko dugu, Views/Ordaindu/Osatu.cshtml

@model Wineshop.ViewModels.OsatuViewModel

```


@{
    ViewData["Title"] = "Osatu";
}

<div class="container">
    <p>@Html.DisplayFor(modelItem => Model.Bezeroa), zure erosketa ondo osatu da</p>
    <table class="table">
        <thead>
            <tr>
                <th>
                    Ardoa
                </th>
                <th>
                    Kantitatea
                </th>
            </tr>
        </thead>
        <tbody>
            @foreach (var item in Model.SaskiaAleak)
            {
                <tr>
                    <td>
                        @Html.DisplayFor(modelItem => item.Ardoald)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.Kantitatea)
                    </td>
                </tr>
            }
            <tr>
                <td>
                    <a class="btn btn-borgona text-light" asp-controller="Home" asp-action="Index">Bueltatu</a>
                    <a class="btn btn-borgona text-light" asp-controller="Report" asp-action="Inprimatu">Tiketa</a>
                </td>
                <td></td>
            </tr>
        </tbody>
    </table>
</div>

```

WineShop web aplikazioa

Home Privacy Hello izaskun@gmail.com!



Finest Wine Selection - ardoaonline.com

Izaskun Ni, zure erosketa ondo osatu da

Ardoa	Kantitatea
2	2
1	1

Ikusten den bezala, ardoaren Id-a bistaratzen da.

**** Behar diren aldaketa guztiak egin, ardoaren izena bistaratzeko.**

**** Pista: begiratu SaskiaController- Index metodoa.**

WineShop web aplikazioa**OsatuViewModel klasean aldaketa egingo dugu:**

```
public class OsatuViewModel
{
    public string Bezeroa { get; set; }
    public string Saskiald { get; set; }
    public IList<SaskiaAleaViewModel> SaskiaAleaVMList { get; set; }
}
```

Eta Kontroladorean, Osatu metodoan:**(ArdoaService zerbitzuak konstruktorean injektatu)**

```
public async Task<IActionResult> Osatu(string bezeroa, string saskiald)
{
    .....
```

```
IList<SaskiaAlea> saskiaAleaList = new List<SaskiaAlea>();
saskiaAleaList = await _saskiaService.SaskiaLortuAleak(id);
```

```
//Ardo bakoitzaren datuak hartu eta ViewModel bezala sortu
```

```
IList<SaskiaAleaViewModel> saskiaAleaVMList = new List<SaskiaAleaViewModel>();
```

```
foreach (var saskiaAlea in saskiaAleaList)
```

```
{
    var ardoa = await _ardoaService.GetArdoa(saskiaAlea.Ardoald);
    SaskiaAleaViewModel saskiaAleaViewModel = new SaskiaAleaViewModel()
    {
        Ardoald = ardoa.Id,
        Irudia = ardoa.Irudia,
        Izena = ardoa.Izena,
        Kantitatea = saskiaAlea.Kantitatea,
        Salneurria = ardoa.Salneurria
    };
    saskiaAleaVMList.Add(saskiaAleaViewModel);
}
```

```
osatuViewModel.SaskiaAleaVMList = saskiaAleaVMList;
```

```
.....
```

```
}
```

Osatu.cshtml

```
@foreach (var item in Model.SaskiaAleaVMList)
{
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Izena)
        </td>
```

WineShop web aplikazioa

Ajax erabilita karritotik kentzeko:

```

$("#RemoveLink").click(function (e) {
    e.preventDefault();
    // Get the id from the link
    var recordToDel = $(this).attr("data-id");
    var kantitatea = parseInt($(this).attr("data-kantitatea"));
    var salneurria = parseFloat($(this).attr("data-salneurria"));
    var guztira = parseFloat($("#cart-total").text());
    if (recordToDel !== "") {
        // Perform the ajax post
        $.post("/Saskia/SaskiaKenduAjax", { "id": recordToDel, "kantitatea": kantitatea, "salneurria": salneurria, "guztira": guztira },
            function (data) {
                // Successful requests get here
                // Update the page elements
                $("#item-count-" + recordToDel).text(data.kantitatea.toString());
                $("#cart-total").text((data.guztira + data.salneurria).toString());
                $("#update-message").text(data.mezua);
            });
    }
});

```

SaskiaController.cs

```

public async Task<ActionResult> SaskiaKenduAjax(int id, int kantitatea, float salneurria, float guztira)
{
    var cart = Saskia.SaskiaLortu(this.HttpContext);
    await _saskiaService.SaskiaKendu(id, cart.Saskiald);
    kantitatea--;
    var results = new
    {
        mezua = "Zure saskia eguneratu da",
        kantitatea = kantitatea,
        salneurria = salneurria,
        guztira = guztira
    };
    return Json(results);
}

```

WineShop web aplikazioa**SaskiaService:**

```
public async Task SaskiaKendu(int ardoald, string saskiald)
{
    Uri rutasaskia = new Uri(rutaTodos, saskiald);

    List<SaskiaAlea> saskiaAleaList = new List<SaskiaAlea>();
    saskiaAleaList = await SaskiaLortuAleak(saskiald);

    SaskiaAlea cartitem = new SaskiaAlea();
    cartitem = saskiaAleaList.FirstOrDefault(s => s.Ardoald == ardoald);

    cartitem.Kantitatea--;
    using (var httpClient = new HttpClient())
    {
        StringContent content = new StringContent(JsonConvert.SerializeObject(cartitem), Encoding.UTF8, "application/json");
        var response = await httpClient.PutAsync(rutasaskia, content);
        response.EnsureSuccessStatusCode();
    }
}
```

Zer egin behar da kantitatea 0 bada?

WineShop web aplikazioa

Karritoa bistaratzeko ViewComponent bat sortuko dugu

