T.C.
ÇUKUROVA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
STAJ DEFTERİ

ÖĞRENCİNİN	Bölümü	Bilgisayar Mühendisliği	
	Numarası	2018556301	
	Adı, Soyadı	MUHAMMED NUR ALDIN HALUBİ	
	Okula Girdiği Yıl	2018	
	Bu Çalışmayı Kaçıncı Devre Pratik Çalışması Olarak Yaptığı?	1. Devre □	2. Devre ☒
	Pratik Çalışmaya Başladığı Tarih	20.0[illegible]/2022	
	Pratik Çalışmayı Bitirdiği Tarih	0[illegible]/2022	
	Kaç İş Günü Pratik Çalışma Yaptığı	20	

T.C. ÇUKUROVA ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ DEKANLIĞI

Yanında Pratik Çalışma Yapılan	İşyeri veya Firmanın Adı ve Adresi		
	İşyeri veya Firmanın adına defteri tasdik eden işyeri amirinin:		Yukarıda ismi ve fotoğrafı bulunan öğrencinin iş yerimizde .... iş günü Pratik Çalışma yaptğını ve bu defteri kendisinin tanzim ettiğini beyan ve tastik ederim .../.../20..
	Ünvanı	Bilgisayar Mühendisi	
	Adı	:Kaan Kandemir	İmza ve Mühür Kaan KANDEMİR Bilgi İşlem
	Diploma no	:639/189	
	Oda sicil no	:..................	

PRATİK ÇALIŞMALARI İNCELEME KOMİSYONUNUN KANAATİ

Yapılan pratik çalışma .... iş günlük ...... devre çalışması olarak kabul Edilmiştir/Edilmemiştir	
Tarih	İsim ve İmza :..
	İsim ve İmza :..
	İsim ve İmza :..

Bu belge Ç.Ü. Bilgisayar Mühendisliği bölümünden **Doç. Dr. Zekeriya TÜFEKÇİ** tarafından **"UYGUN"** bulunup onaylanmıştır.
Doğrulama kodu: 621e2e2b2f247
Doğrulama linki: https://ceng.cu.edu.tr/belge.php?kod=621e2e2b2f247

PRATİK ÇALIŞMANIN GÜNLERE DAĞILMA ÇİZELGESİ

Öğrencinin Çalıştığı Günler	Günlük Çalışma Saati	İşyeri Amirinin İmzası	Öğrencinin Çalıştığı Konular
01/03/2022	8	Kaan KANDEMİR Bilgi İşlem	Introduction and set-ups
02/03/2022	8		Some BasicSql command and their usage
03/03/2022	8		Sql Functios . Procedures
04/03/2022	8		View Sctructure and Join
07/03/2022	8	Kaan KANDEMİR Bilgi İşlem	Building nav-bar with html-css
08/03/2022	8		Product repository for CRUD operations
09/03/2022	8		Completed the product repository
10/03/2022	8		Products controller with adding and getting endpoints.
11/03/2022	8		Adding the delete and update methods in controller
14/03/2022	8	Kaan KANDEMİR Bilgi İşlem	
15/03/2022	8		Added photos to products with relationships
16/03/2022	8		Created a new shiny photo service! :D
17/03/2022	8		Attended event organized by the company
18/03/2022	8	Kaan KANDEMİR Bilgi İşlem	Created method for adding photos to products
21/03/2022	8		Created more end-points for photos.
22/03/2022	8		Created new from for adding products.
23/03/2022	8		Created product details component.
24/03/2022	8		Created photo uploader
25/03/2022	8	Kaan KANDEMİR Bilgi İşlem	Making the HTML part of the photo uploader.
28/03/2022	8		Adding product DTO and mapping the products to it

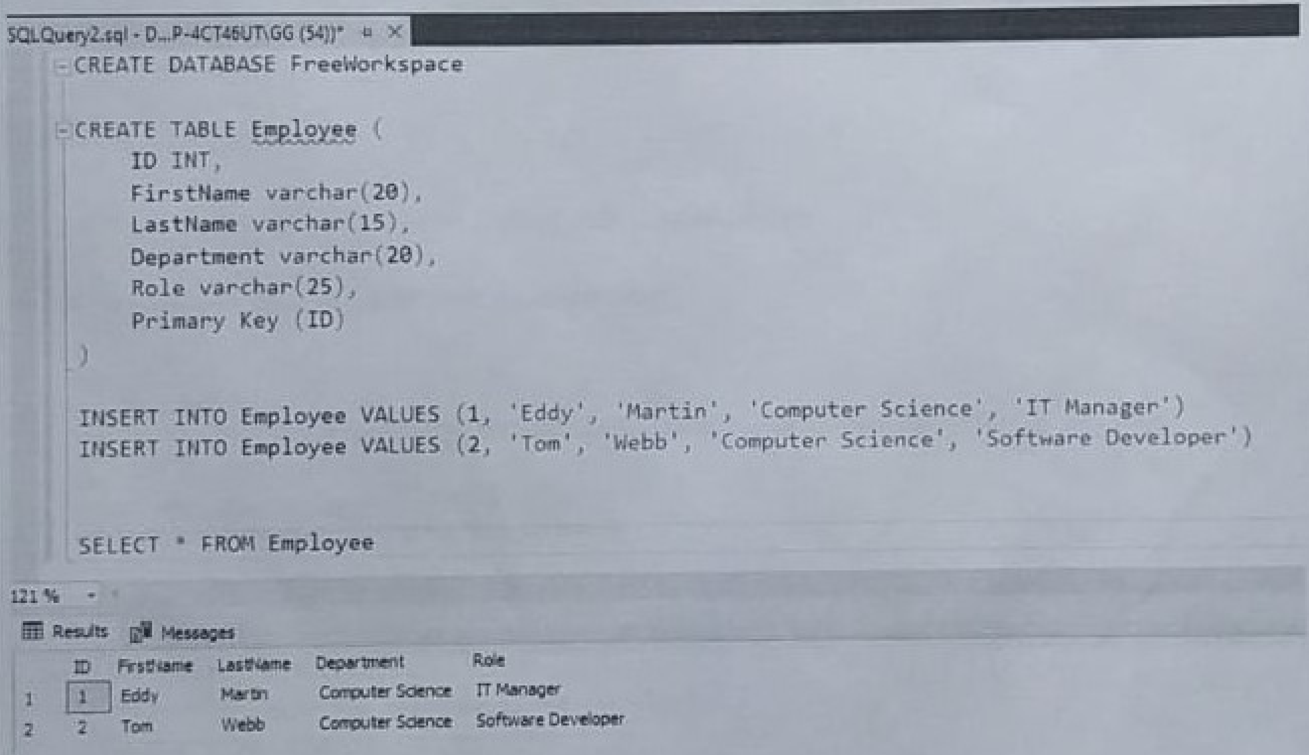
Today is my first day of the internship. I met my supervisers and checked around the different departments of the company, then we discussed the curriculum along with the topics that I'll be learning and decided to continue with Web Development and learning technologies used in it.

After being done with discussions and introductions I set the pc that I'll be using in the internsship and got all applications and add-ons like Vs Code, Postman, Angular extensions, etc.. installed to start working on the next day.

Sayfa No	Çalışmanın		KONTROL
1	Konusu :Introduction and set-ups	Yapıldığı Tarih 01/03/2022	Kaan KANDEMİR Bilgi İşlem

Today I started with SQL and checked on the basic commands like CREATE TABLE, CREATE DATABASE, SELECT FROM, etc... and had a general understanding of tables and how they work.

Some of the used codes:



```
CREATE DATABASE FreeWorkspace

CREATE TABLE Employee (
    ID INT,
    FirstName varchar(20),
    LastName varchar(15),
    Department varchar(20),
    Role varchar(25),
    Primary Key (ID)
)

INSERT INTO Employee VALUES (1, 'Eddy', 'Martin', 'Computer Science', 'IT Manager')
INSERT INTO Employee VALUES (2, 'Tom', 'Webb', 'Computer Science', 'Software Developer')


SELECT * FROM Employee
```

```
   ID  FirstName  LastName  Department        Role
1  1   Eddy       Martin    Computer Science  IT Manager
2  2   Tom        Webb      Computer Science  Software Developer
```

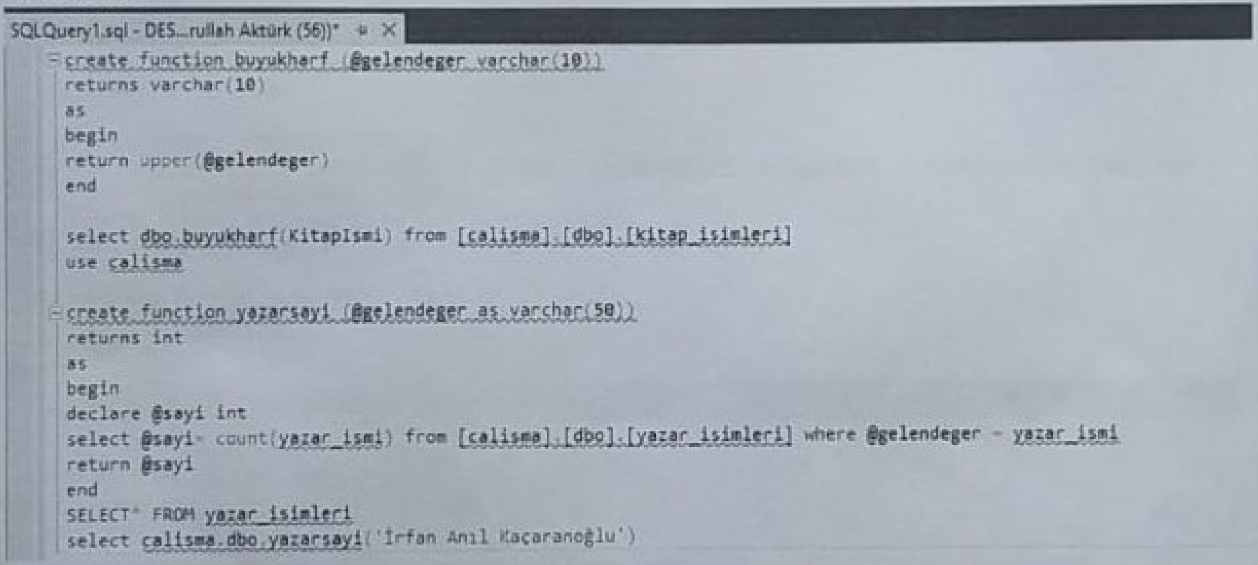
Sayfa No	Çalışmanın		KONTROL
2	Konusu : Some basic SQL commands and their usage.	Yapıldığı Tarih 02/03/2022	.... Kazım KANDEMİR Bilgi İşlem

Today I looked Procedure and Function structures in SQL and made some practice

Examples on them.

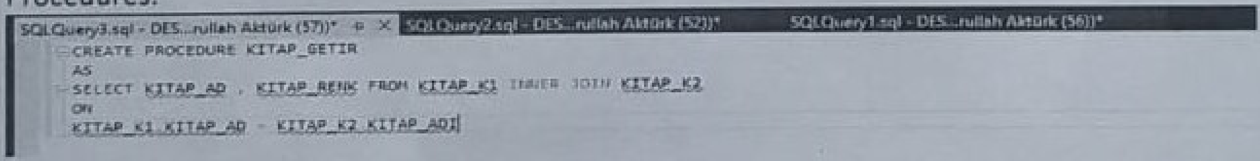
Some of the used codes:

Functions:



```
create function buyukharf (@gelendeger varchar(10))
returns varchar(10)
as
begin
return upper(@gelendeger)
end

select dbo.buyukharf(KitapIsmi) from [calisma].[dbo].[kitap_isimleri]
use calisma

create function yazarsayi (@gelendeger as varchar(50))
returns int
as
begin
declare @sayi int
select @sayi= count(yazar_ismi) from [calisma].[dbo].[yazar_isimleri] where @gelendeger = yazar_ismi
return @sayi
end
SELECT* FROM yazar_isimleri
select calisma.dbo.yazarsayi('İrfan Anil Kaçaranoğlu')
```

Procedures:


SQLQuery3.sql - DES...rullah Aktürk (57))*
SQLQuery2.sql - DES...rullah Aktürk (52))*
SQLQuery1.sql - DES...rullah Aktürk (56))*

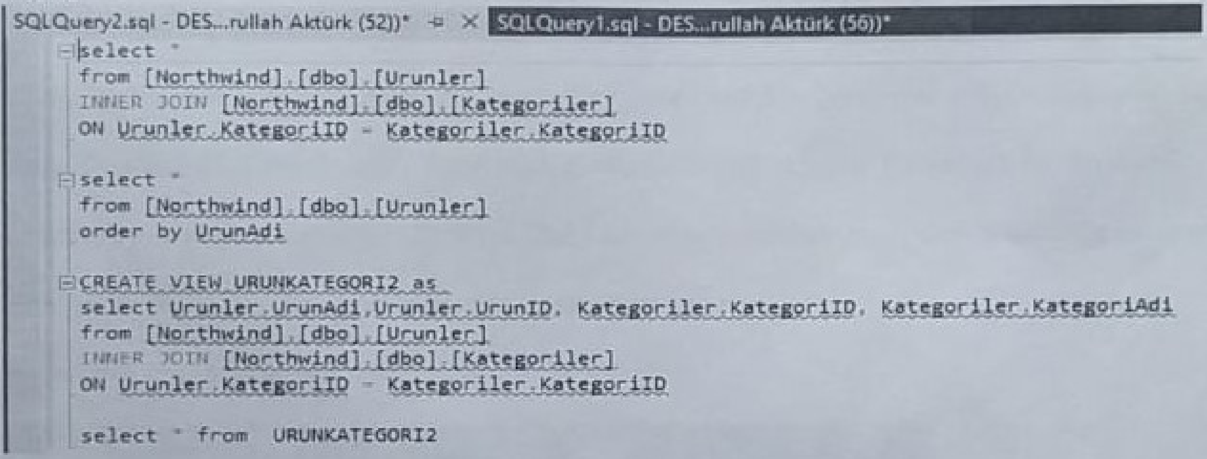
```
CREATE PROCEDURE KITAP_GETIR
AS
SELECT KITAP_AD , KITAP_RENK FROM KITAP_K1 INNER JOIN KITAP_K2
ON
KITAP_K1.KITAP_AD = KITAP_K2.KITAP_ADI
```

Sayfa No	Çalışmanın		KONTROL
3	Konusu :SQL Functuions, Procedures	Yapıldığı Tarih 03/03/2022	.... Kaan KANDEMİR Bilgi İşlem

Today I learned the View structure and Join types (Inner, Outer, Full, etc..) and made some practice examples.

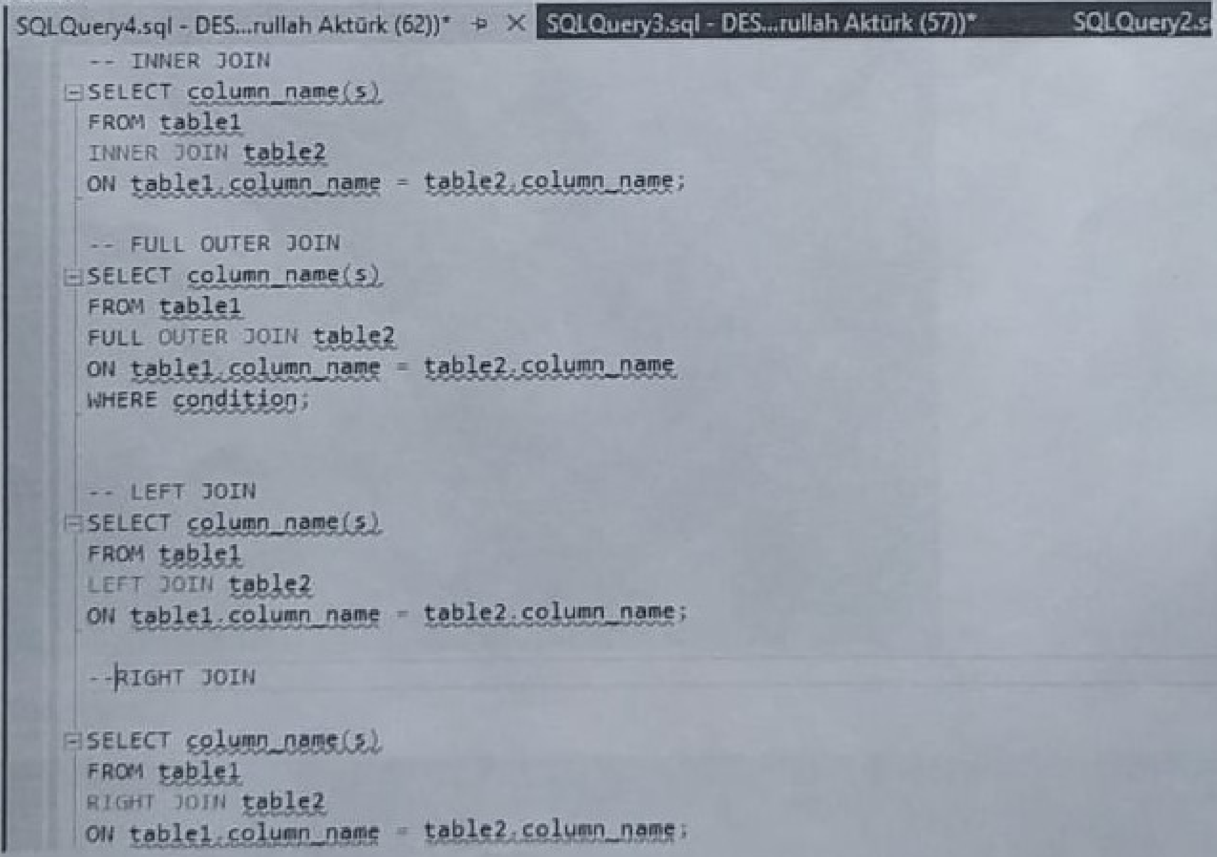
Some of the used codes:

View:


SQLQuery2.sql - DES...rullah Aktürk (52))*
SQLQuery1.sql - DES...rullah Aktürk (56))*

```
select *
from [Northwind].[dbo].[Urunler]
INNER JOIN [Northwind].[dbo].[Kategoriler]
ON Urunler.KategoriID = Kategoriler.KategoriID

select *
from [Northwind].[dbo].[Urunler]
order by UrunAdi

CREATE VIEW URUNKATEGORI2 as
select Urunler.UrunAdi,Urunler.UrunID, Kategoriler.KategoriID, Kategoriler.KategoriAdi
from [Northwind].[dbo].[Urunler]
INNER JOIN [Northwind].[dbo].[Kategoriler]
ON Urunler.KategoriID = Kategoriler.KategoriID

select * from  URUNKATEGORI2
```

Joins:

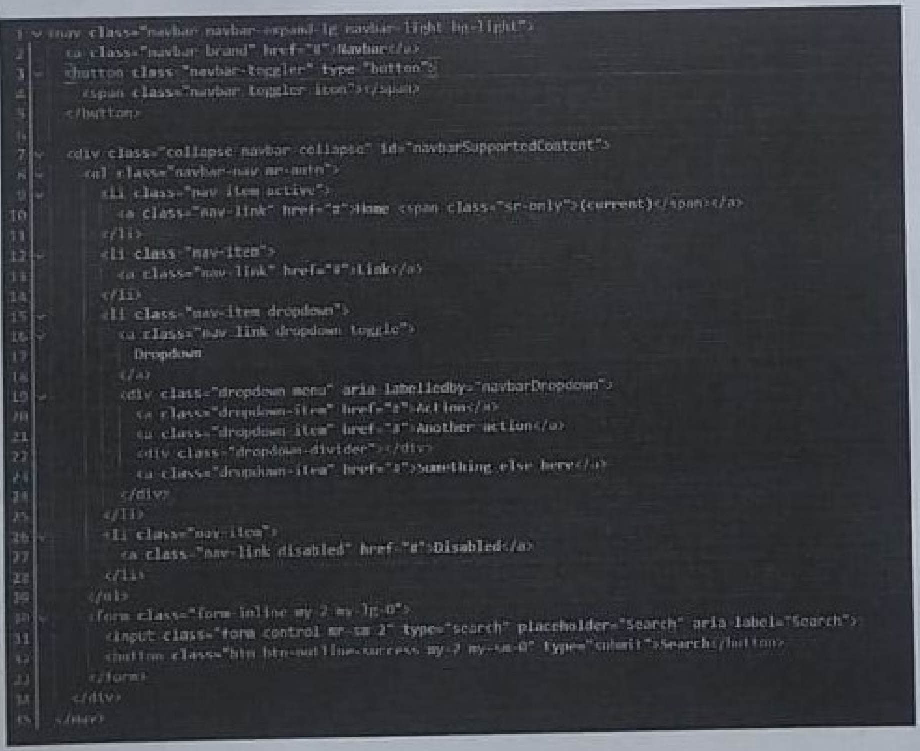

SQLQuery4.sql - DES...rullah Aktürk (62))*
SQLQuery3.sql - DES...rullah Aktürk (57))*
SQLQuery2.s

```
 -- INNER JOIN
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;

 -- FULL OUTER JOIN
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;


 -- LEFT JOIN
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;

 --RIGHT JOIN

SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

Sayfa No	Çalışmanın		KONTROL
4	Konusu :View structure and Joins in SQL.	Yapıldığı Tarih 04/03/2022	.... [illegible] KANDEMİR Bilgi İşlem

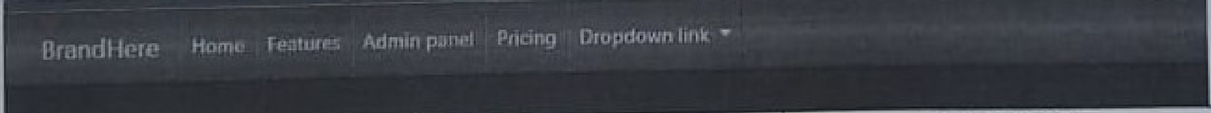
Today after finishing the SQL basics I consulted my advisors about what to do next, they told me that I should start developing some application or real world project so I asked them if I could continue working on a project that I had started developing on my won and they agreed. The topic of the project is an e-commerce website for selling goods which is developed with C# Asp.net core for back-end, SQL lite with entity framework for database and, Angular framework for front-end. I started by setting up the project on the pc that I am using in the company, then created the nav-bar component hat will contain the navigation links of the website.

Some of the used codes:



```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle">
            Dropdown
          </a>
          <div class="dropdown-menu" aria-labelledby="navbarDropdown">
            <a class="dropdown-item" href="#">Action</a>
            <a class="dropdown-item" href="#">Another action</a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item" href="#">Something else here</a>
          </div>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" href="#">Disabled</a>
        </li>
      </ul>
      <form class="form-inline my-2 my-lg-0">
        <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
        <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
      </form>
    </div>
  </nav>
```

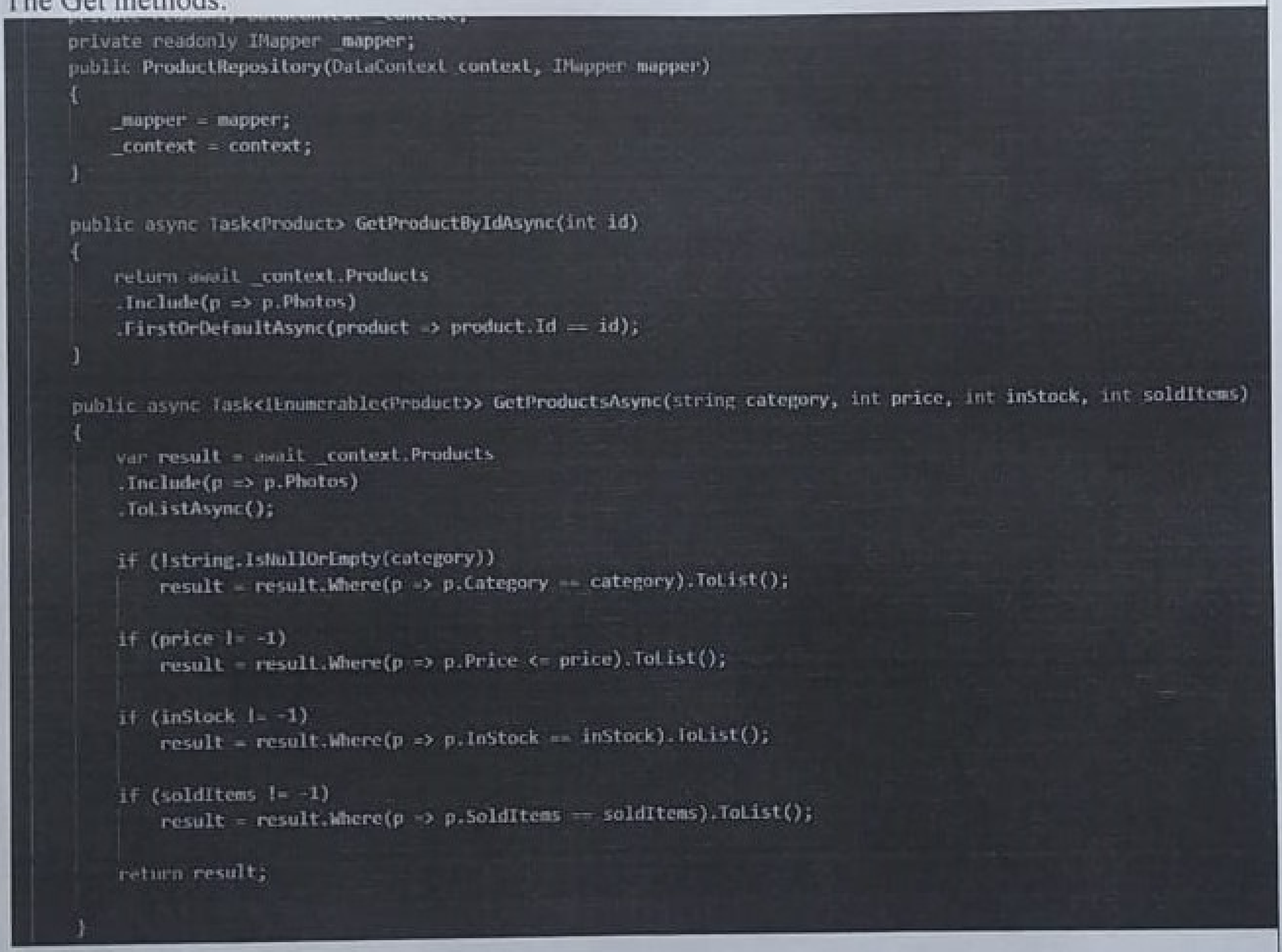
The output:



Sayfa No	Çalışmanın		KONTROL
5	Konusu : Building nav bar with HTML and CSS	Yapıldığı Tarih 07/03/2022	....Kaan KANDEMİR Bilgi İşlem

Today I started working on the back-end I added a new model for products, then created a Repository for dealing with adding deleting, and getting products from the database, For now the Repo has only methods for getting products, But I will add more methods tommorow.

The Get methods:

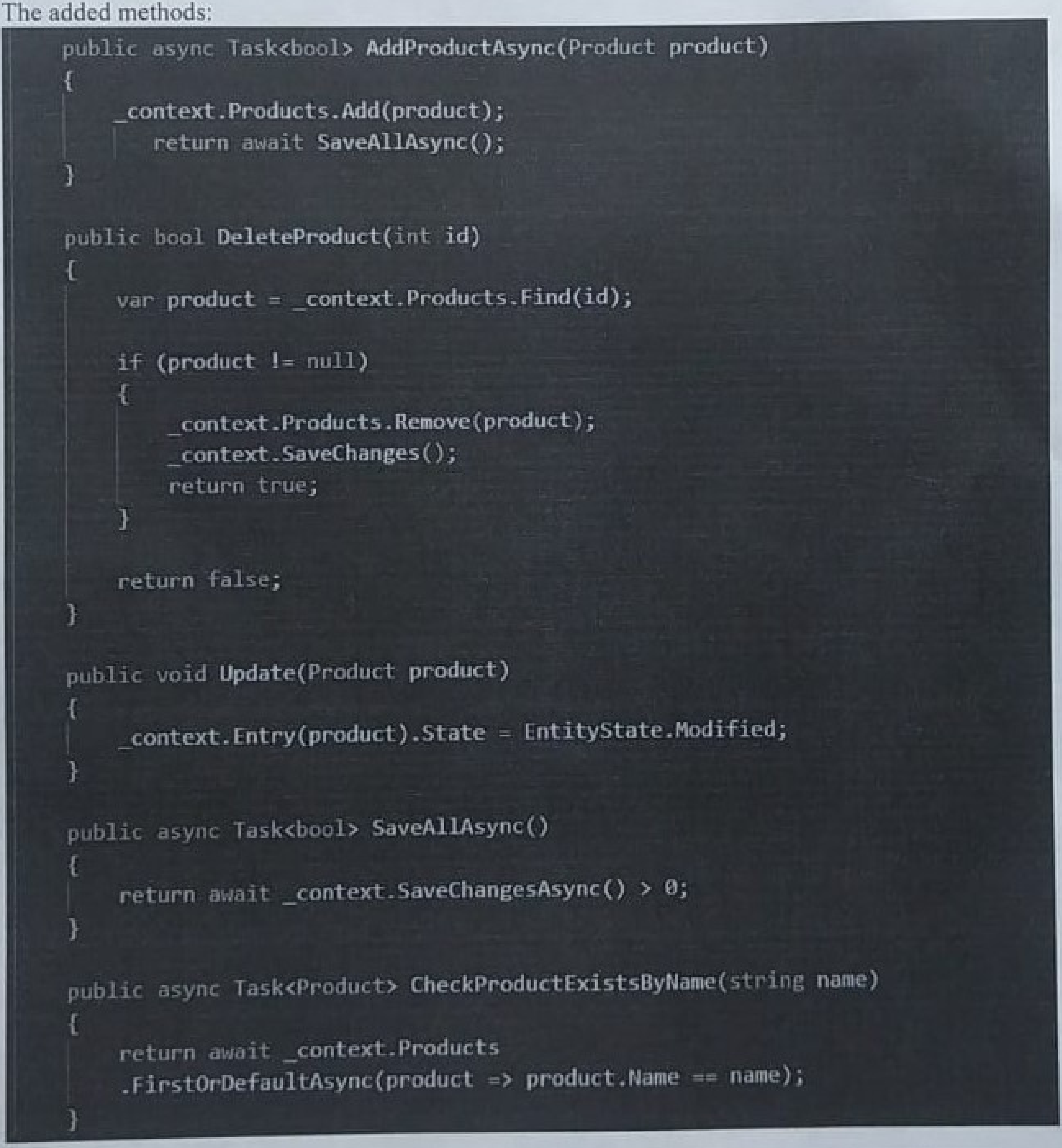


```
    private readonly IMapper _mapper;
    public ProductRepository(DataContext context, IMapper mapper)
    {
        _mapper = mapper;
        _context = context;
    }

    public async Task<Product> GetProductByIdAsync(int id)
    {
        return await _context.Products
        .Include(p => p.Photos)
        .FirstOrDefaultAsync(product => product.Id == id);
    }

    public async Task<IEnumerable<Product>> GetProductsAsync(string category, int price, int inStock, int soldItems)
    {
        var result = await _context.Products
        .Include(p => p.Photos)
        .ToListAsync();

        if (!string.IsNullOrEmpty(category))
            result = result.Where(p => p.Category == category).ToList();

        if (price != -1)
            result = result.Where(p => p.Price <= price).ToList();

        if (inStock != -1)
            result = result.Where(p => p.InStock == inStock).ToList();

        if (soldItems != -1)
            result = result.Where(p => p.SoldItems == soldItems).ToList();

        return result;

    }
```

Sayfa No	Çalışmanın		KONTROL
6	Konusu : Product repository for CRUD operations.	Yapıldığı Tarih 08/03/2022	.... Kaan KANDEMİR Bilgi İşlem

Today I continued working on the product repository from where I stopped yesterday and added the rest of the CRUD methods with some helper methods for persisting the changes on the database and checking whether a name of product exists to prevent having duplicate names for different products.

The added methods:

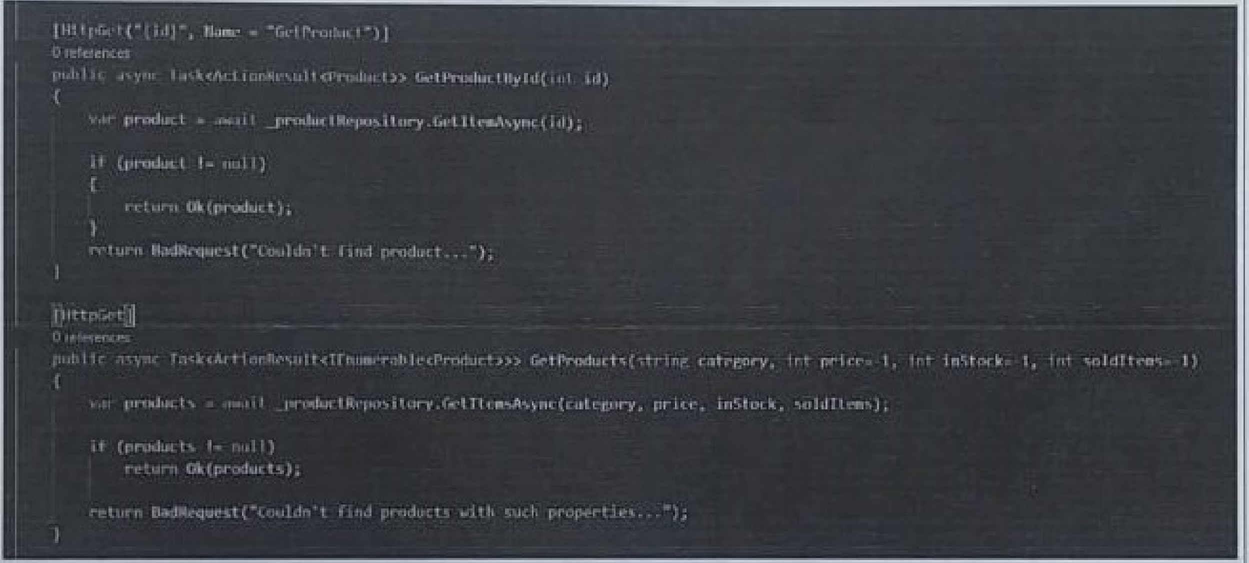


```
public async Task<bool> AddProductAsync(Product product)
{
    _context.Products.Add(product);
    return await SaveAllAsync();
}

public bool DeleteProduct(int id)
{
    var product = _context.Products.Find(id);

    if (product != null)
    {
        _context.Products.Remove(product);
        _context.SaveChanges();
        return true;
    }

    return false;
}

public void Update(Product product)
{
    _context.Entry(product).State = EntityState.Modified;
}

public async Task<bool> SaveAllAsync()
{
    return await _context.SaveChangesAsync() > 0;
}

public async Task<Product> CheckProductExistsByName(string name)
{
    return await _context.Products
    .FirstOrDefaultAsync(product => product.Name == name);
}
```

Sayfa No	Çalışmanın		KONTROL
7	Konusu: Completed the product repository.	Yapıldığı Tarih 09/03/2022	.... KANDEMİR Bilgi İşlem

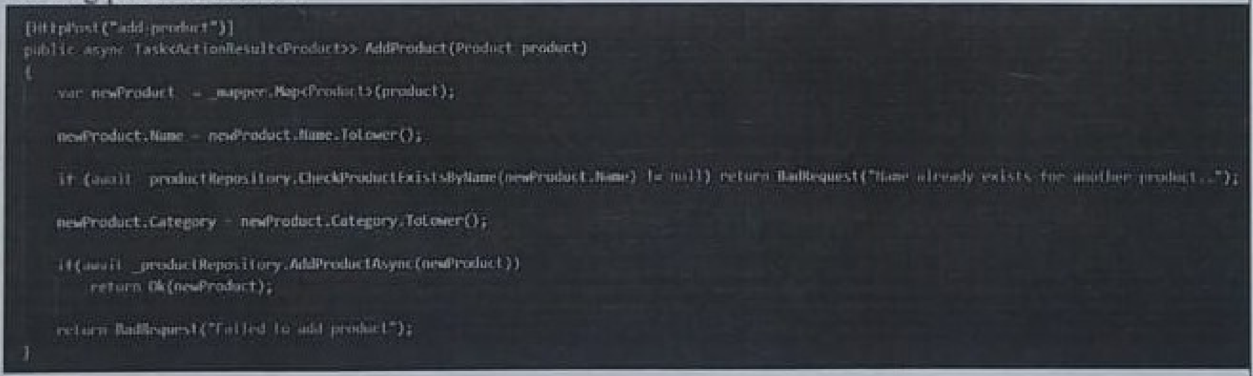
Today I created a controller that contains two methods for getting products, and another method for adding new products and tested them using Postman.

The Get methods:



```
[HttpGet("{id}", Name = "GetProduct")]
0 references
public async Task<ActionResult<Product>> GetProductById(int id)
{
    var product = await _productRepository.GetItemAsync(id);

    if (product != null)
    {
        return Ok(product);
    }
    return BadRequest("Couldn't find product...");
}

[HttpGet]
0 references
public async Task<ActionResult<IEnumerable<Product>>> GetProducts(string category, int price=-1, int inStock=-1, int soldItems=-1)
{
    var products = await _productRepository.GetItemsAsync(category, price, inStock, soldItems);

    if (products != null)
        return Ok(products);

    return BadRequest("Couldn't find products with such properties...");
}
```

Adding product method:

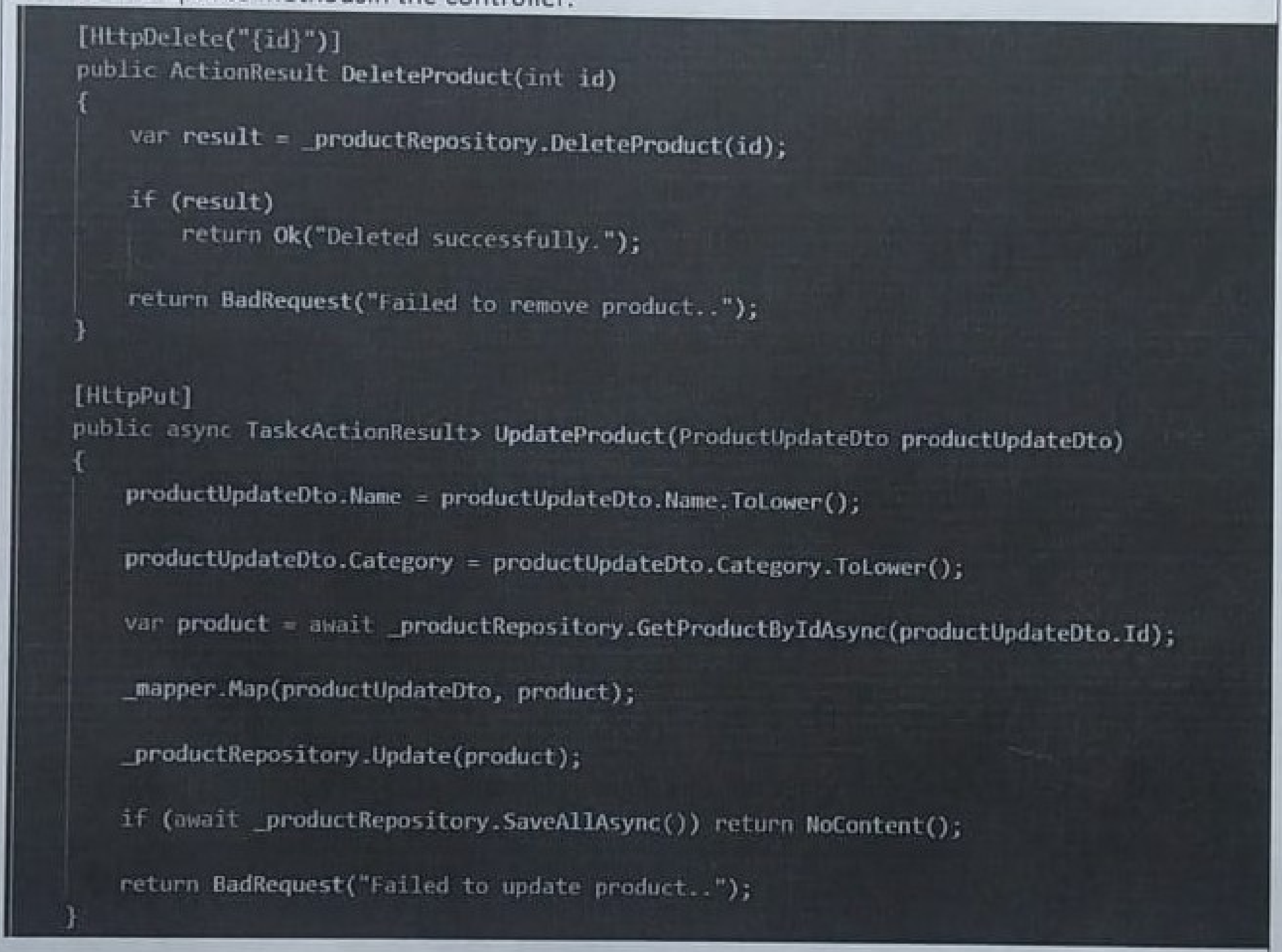


```
[HttpPost("add-product")]
public async Task<ActionResult<Product>> AddProduct(Product product)
{
    var newProduct  = _mapper.Map<Product>(product);

    newProduct.Name = newProduct.Name.ToLower();

    if (await _productRepository.CheckProductExistsByName(newProduct.Name) != null) return BadRequest("Name already exists for another product..");

    newProduct.Category = newProduct.Category.ToLower();

    if(await _productRepository.AddProductAsync(newProduct))
        return Ok(newProduct);

    return BadRequest("Failed to add product");
}
```

Sayfa No	Çalışmanın		KONTROL
8	Konusu: Products controller with adding and getting endpoints.	Yapıldığı Tarih 10/03/2022	.... aan KANDEMIR Bilgi İşlem

Today I continued working on the products controller and added the two remaining two CRUD funtionalities which are delete and update.

Delete and update methodsin the controller:

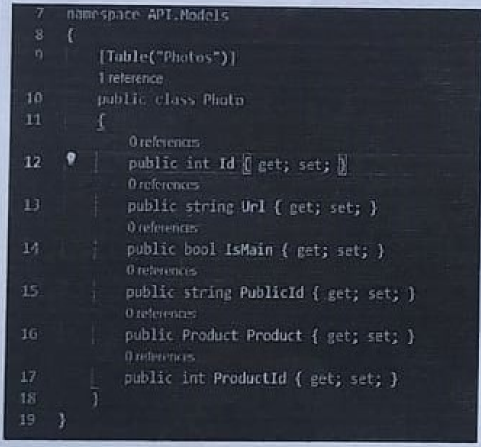


```
[HttpDelete("{id}")]
public ActionResult DeleteProduct(int id)
{
    var result = _productRepository.DeleteProduct(id);

    if (result)
        return Ok("Deleted successfully.");

    return BadRequest("Failed to remove product..");
}

[HttpPut]
public async Task<ActionResult> UpdateProduct(ProductUpdateDto productUpdateDto)
{
    productUpdateDto.Name = productUpdateDto.Name.ToLower();

    productUpdateDto.Category = productUpdateDto.Category.ToLower();

    var product = await _productRepository.GetProductByIdAsync(productUpdateDto.Id);

    _mapper.Map(productUpdateDto, product);

    _productRepository.Update(product);

    if (await _productRepository.SaveAllAsync()) return NoContent();

    return BadRequest("Failed to update product..");
}
```

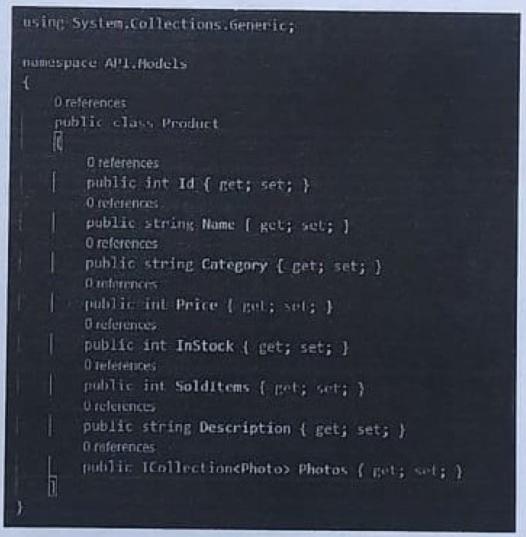
Sayfa No	Çalışmanın		KONTROL
9	Konusu: Adding the delete and update methods in controller.	Yapıldığı Tarih 11/03/2022	.... Kaan KANDEMIR Bilgi İşlem

Today I wanted to add a collection of photos to my products but I didn't know how so I spent most of my time watching videos and reading articles about Entity Framework relationships and how they work and coded properly, and of course with some help from my advisors I managed to get a basic understanding of how these relationships work. Afterwards I applied what I learned on the new Photo model and Product model and estableshed the relationship between them as one to many because one product can have multiple photos while each photo can have only one product. And apparently all I had to do was to add a collection of photos to the Product model and a single product to the Photo model and Entity Framework core would handle the rest.

The models and changes they got



```
namespace API.Models
{
    [Table("Photos")]
    1 reference
    public class Photo
    {
        0 references
        public int Id { get; set; }
        0 references
        public string Url { get; set; }
        0 references
        public bool IsMain { get; set; }
        0 references
        public string PublicId { get; set; }
        0 references
        public Product Product { get; set; }
        0 references
        public int ProductId { get; set; }
    }
}
```

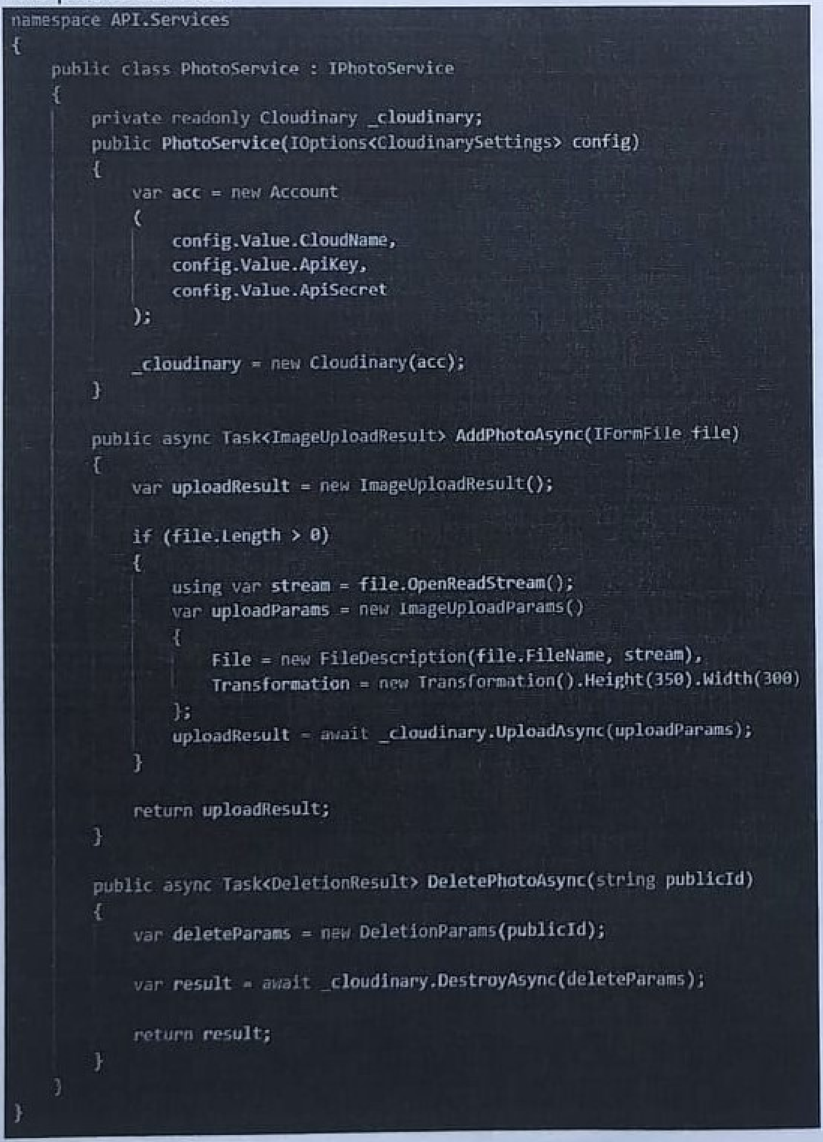


```
using System.Collections.Generic;

namespace API.Models
{
    0 references
    public class Product
    {
        0 references
        public int Id { get; set; }
        0 references
        public string Name [ get; set; ]
        0 references
        public string Category { get; set; }
        0 references
        public int Price { get; set; }
        0 references
        public int InStock { get; set; }
        0 references
        public int SoldItems { get; set; }
        0 references
        public string Description { get; set; }
        0 references
        public ICollection<Photo> Photos { get; set; }
    }
}
```

Sayfa No	Çalışmanın		KONTROL
10	Konusu: Added photos to products with relationships.	Yapıldığı Tarih 14/03/2022	....Kaan KANDEMİR Bilgi İşlem

Today I added a new service for uploading the photos into some cloud to store it there instead of storing them locally on my machine. The cloud that I chose was Cloudinary and luckily it offers couple of GBs of storage for free before having to buy a premium account. Even though it is quite straight forward it was hard to configure it so it did take me a while, but I managed to make it at the end.

The photo service:

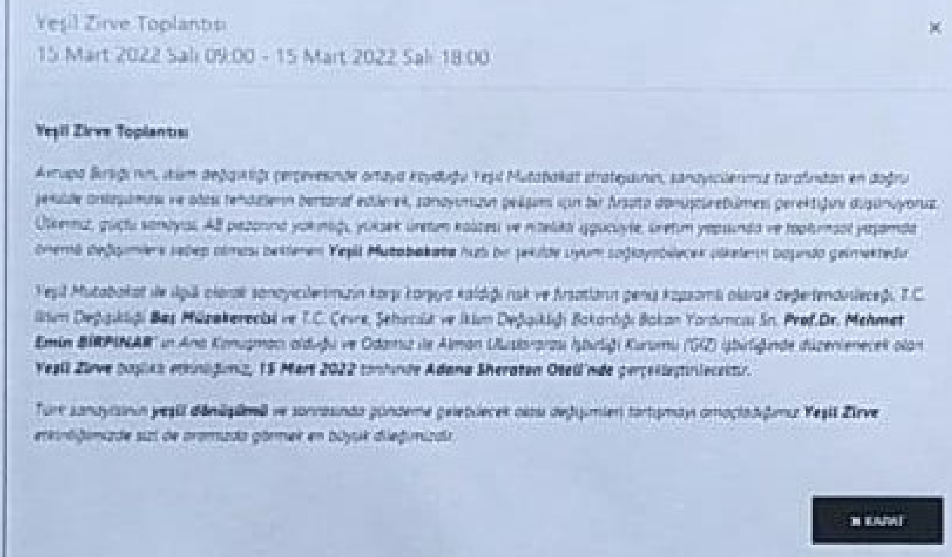


```
namespace API.Services
{
    public class PhotoService : IPhotoService
    {
        private readonly Cloudinary _cloudinary;
        public PhotoService(IOptions<CloudinarySettings> config)
        {
            var acc = new Account
            (
                config.Value.CloudName,
                config.Value.ApiKey,
                config.Value.ApiSecret
            );

            _cloudinary = new Cloudinary(acc);
        }

        public async Task<ImageUploadResult> AddPhotoAsync(IFormFile file)
        {
            var uploadResult = new ImageUploadResult();

            if (file.Length > 0)
            {
                using var stream = file.OpenReadStream();
                var uploadParams = new ImageUploadParams()
                {
                    File = new FileDescription(file.FileName, stream),
                    Transformation = new Transformation().Height(350).Width(300)
                };
                uploadResult = await _cloudinary.UploadAsync(uploadParams);
            }

            return uploadResult;
        }

        public async Task<DeletionResult> DeletePhotoAsync(string publicId)
        {
            var deleteParams = new DeletionParams(publicId);

            var result = await _cloudinary.DestroyAsync(deleteParams);

            return result;
        }
    }
}
```

Sayfa No	Çalışmanın		KONTROL
11	Konusu: Created a new shiny photo service! :D	Yapıldığı Tarih 15/03/2022	....Yaşar KANDEMİR Bilgi İşlem

Away from coding and setting in front of the pc I spent this day in an event that the company Adana chamber of industry (Adana sanayi odasi) where I am working has organized.

The event was "Yeşil Zirve" congress and it took place at the Sheraton Hotel and I participated with the IT team who are my advisors.



Yeşil Zirve Toplantısı

15 Mart 2022 Salı 09:00 - 15 Mart 2022 Salı 18:00

Yeşil Zirve Toplantısı

Avrupa Birliği'nin, iklim değişikliği çerçevesinde ortaya koyduğu Yeşil Mutabakat stratejisinin, sanayicilerimiz tarafından en doğru şekilde anlaşılması ve olası tehditlerin bertaraf edilerek, sanayimizin gelişimi için bir fırsata dönüştürülebilmesi gerektiğini düşünüyoruz. Ülkemiz, güçlü sanayisi, AB pazarına yakınlığı, yüksek üretim kalitesi ve nitelikli işgücüyle, üretim yapısında ve toplumsal yaşamda önemli değişimlere sebep olması beklenen ***Yeşil Mutabakata*** *hızlı bir şekilde uyum sağlayabilecek ülkelerin başında gelmektedir.*

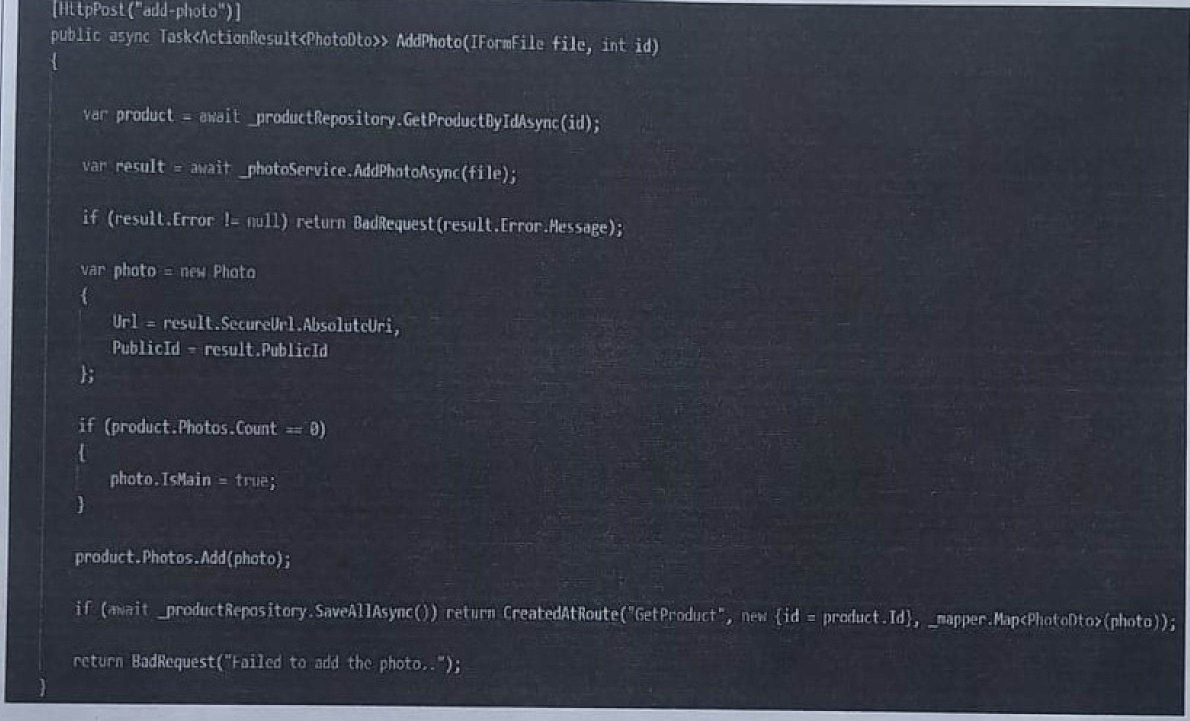
Yeşil Mutabakat ile ilgili olarak sanayicilerimizin karşı karşıya kaldığı risk ve fırsatların geniş kapsamlı olarak değerlendirileceği, T.C. İklim Değişikliği ***Baş Müzakerecisi*** *ve T.C. Çevre, Şehircilik ve İklim Değişikliği Bakanlığı Bakan Yardımcısı Sn.* ***Prof.Dr. Mehmet Emin BİRPINAR****'ın Ana Konuşmacı olduğu ve Odamız ile Alman Uluslararası İşbirliği Kurumu (GIZ) işbirliğinde düzenlenecek olan* ***Yeşil Zirve*** *başlıklı etkinliğimiz,* ***15 Mart 2022*** *tarihinde* ***Adana Sheraton Oteli'nde*** *gerçekleştirilecektir.*

Türk sanayisinin ***yeşil dönüşümü*** *ve sonrasında gündeme gelebilecek olası değişimleri tartışmayı amaçladığımız* ***Yeşil Zirve*** *etkinliğimizde sizi de aramızda görmek en büyük dileğimizdir.*

Sayfa No	Çalışmanın		KONTROL
12	Konusu: Attended event organized by the company.	Yapıldığı Tarih 16/03/2022	.... Kaan KANDEMİR Bilgi İşlem

Today I created one more end-point to the products controller for dealing with photo adding, then I spent the rest of the time toubleshooting and correcting errors that I had in the controller and the photo service that I created before using Postman.

The add photo end-point:

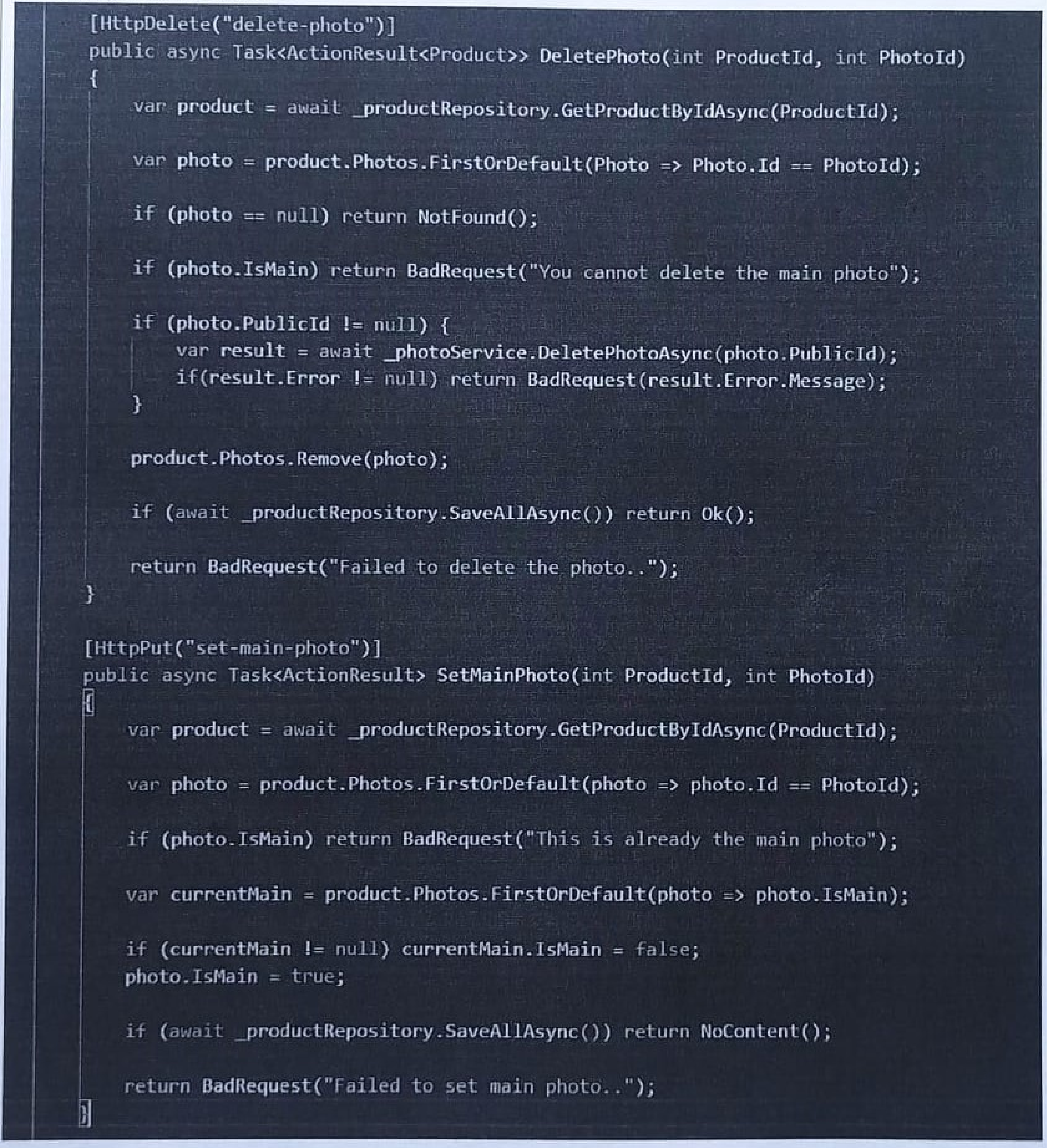


```
[HttpPost("add-photo")]
public async Task<ActionResult<PhotoDto>> AddPhoto(IFormFile file, int id)
{

    var product = await _productRepository.GetProductByIdAsync(id);

    var result = await _photoService.AddPhotoAsync(file);

    if (result.Error != null) return BadRequest(result.Error.Message);

    var photo = new Photo
    {
        Url = result.SecureUrl.AbsoluteUri,
        PublicId = result.PublicId
    };

    if (product.Photos.Count == 0)
    {
        photo.IsMain = true;
    }

    product.Photos.Add(photo);

    if (await _productRepository.SaveAllAsync()) return CreatedAtRoute("GetProduct", new {id = product.Id}, _mapper.Map<PhotoDto>(photo));

    return BadRequest("Failed to add the photo..");
}
```

Sayfa No	Çalışmanın		KONTROL
13	Konusu: Created method for adding photos to products.	Yapıldığı Tarih 17/03/2022	.... Kaan KANDEMİR Bilgi İşlem

Today I completed the products controller by adding two more end-points for deleting and setting a photo as main photo for displaying in the front-end.

The end-points methods:

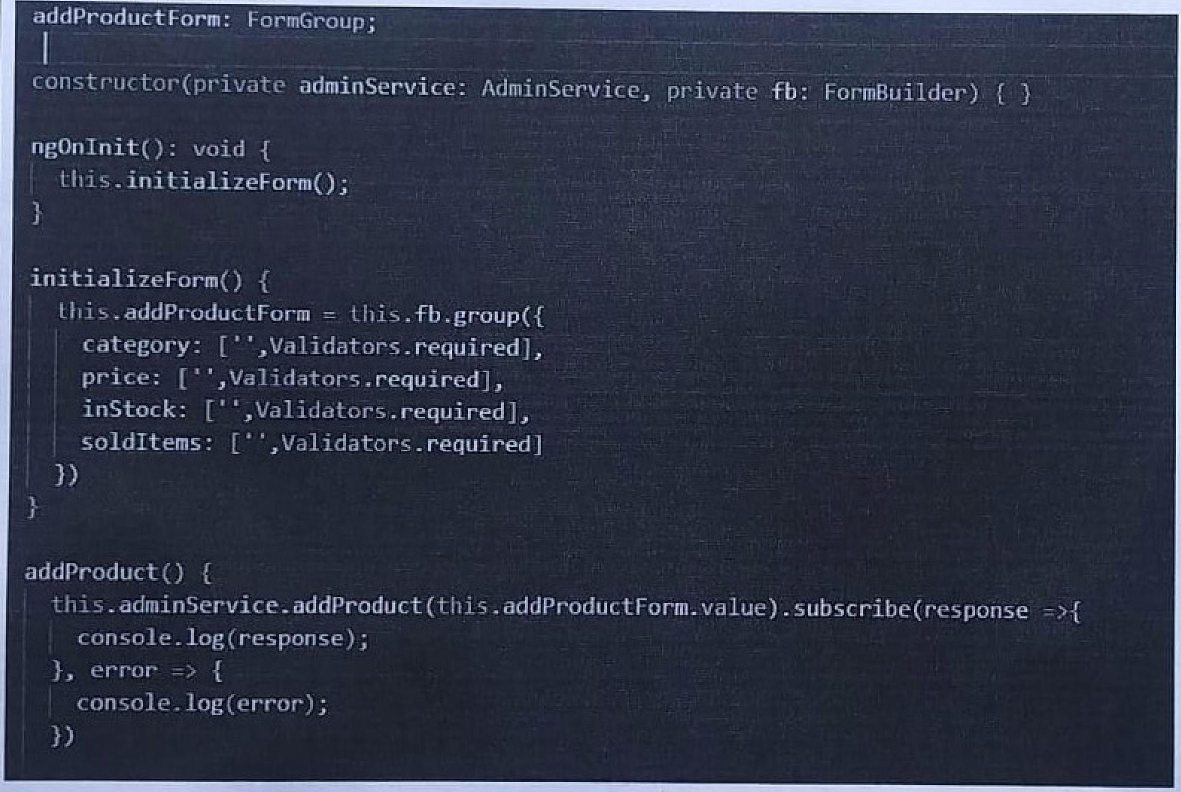


```
[HttpDelete("delete-photo")]
public async Task<ActionResult<Product>> DeletePhoto(int ProductId, int PhotoId)
{
    var product = await _productRepository.GetProductByIdAsync(ProductId);

    var photo = product.Photos.FirstOrDefault(Photo => Photo.Id == PhotoId);

    if (photo == null) return NotFound();

    if (photo.IsMain) return BadRequest("You cannot delete the main photo");

    if (photo.PublicId != null) {
        var result = await _photoService.DeletePhotoAsync(photo.PublicId);
        if(result.Error != null) return BadRequest(result.Error.Message);
    }

    product.Photos.Remove(photo);

    if (await _productRepository.SaveAllAsync()) return Ok();

    return BadRequest("Failed to delete the photo..");
}

[HttpPut("set-main-photo")]
public async Task<ActionResult> SetMainPhoto(int ProductId, int PhotoId)
{
    var product = await _productRepository.GetProductByIdAsync(ProductId);

    var photo = product.Photos.FirstOrDefault(photo => photo.Id == PhotoId);

    if (photo.IsMain) return BadRequest("This is already the main photo");

    var currentMain = product.Photos.FirstOrDefault(photo => photo.IsMain);

    if (currentMain != null) currentMain.IsMain = false;
    photo.IsMain = true;

    if (await _productRepository.SaveAllAsync()) return NoContent();

    return BadRequest("Failed to set main photo..");
}
```

Sayfa No	Çalışmanın		KONTROL
14	Konusu: Created more end-points for photos.	Yapıldığı Tarih 18/03/2022	....Kaan KANDEMİR Bilgi İşlem

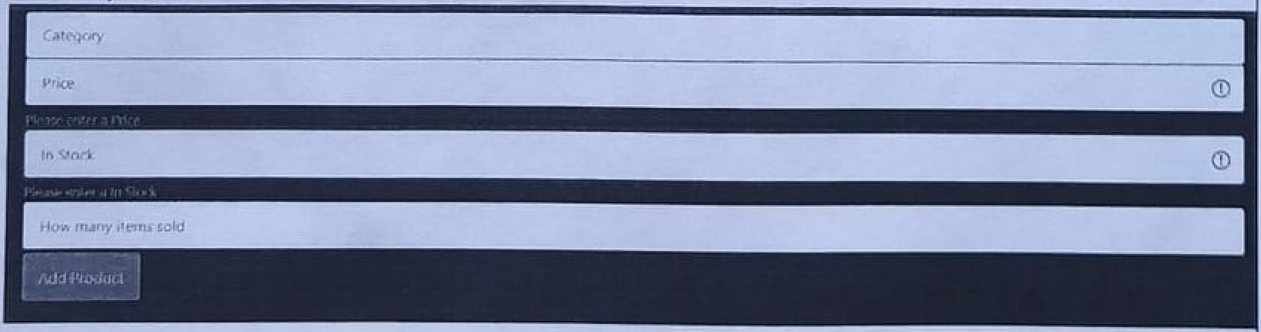
Today I started working on the front-end part of my website and created a component for adding new products then I connected it with the back-end API using http requests. The component applies angular reactive forms and uses the FormBuilder in order to make the form group and form controls within it. Of coure I added some validation to the inputs to make sure nothing goes wrong.

The component code:



```
addProductForm: FormGroup;

constructor(private adminService: AdminService, private fb: FormBuilder) { }

ngOnInit(): void {
  this.initializeForm();
}

initializeForm() {
  this.addProductForm = this.fb.group({
    category: ['',Validators.required],
    price: ['',Validators.required],
    inStock: ['',Validators.required],
    soldItems: ['',Validators.required]
  })
}

addProduct() {
  this.adminService.addProduct(this.addProductForm.value).subscribe(response =>{
    console.log(response);
  }, error => {
    console.log(error);
  })
```

The output:



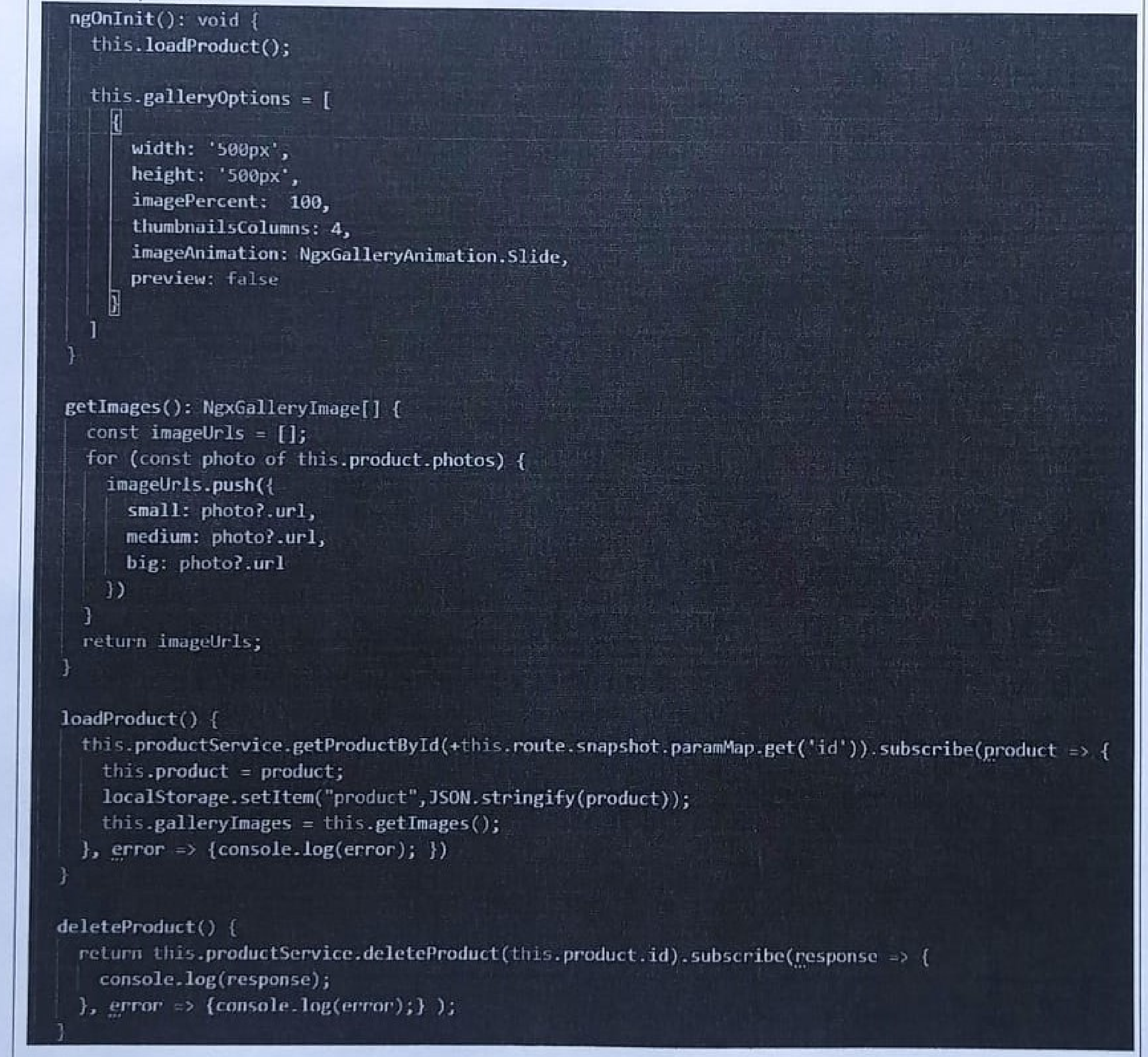
Please enter a Price

Please enter a In Stock

Sayfa No	Çalışmanın		KONTROL
15	Konusu: Created new from for adding products.	Yapıldığı Tarih 21/03/2022	.... Kaan KANDEMİR Bilgi İşlem

Today I created another component for showing details about the product then added two buttons in it, one for deleting the product and the other one for editing it. then made one more component for editing the component and adding more photos on it.

The component:



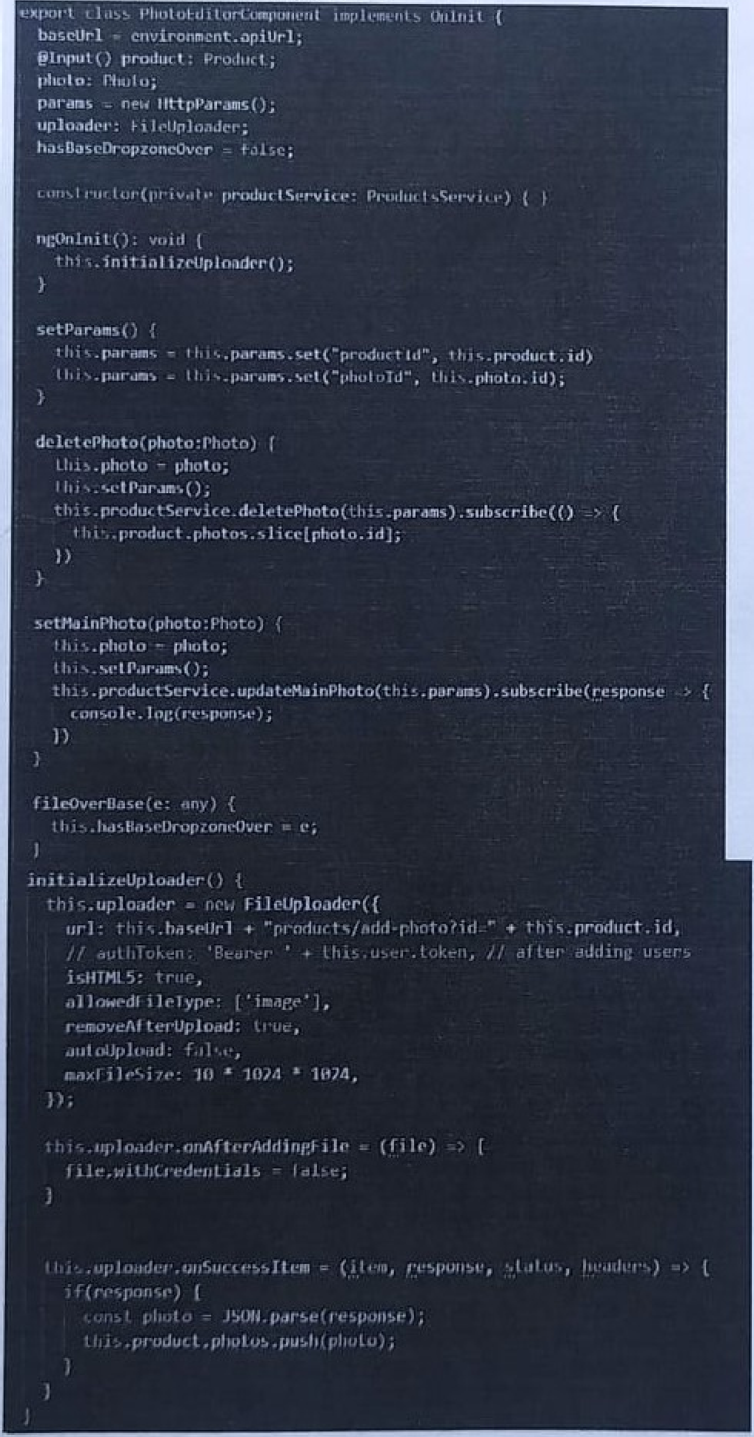
```
ngOnInit(): void {
  this.loadProduct();

  this.galleryOptions = [
    {
      width: '500px',
      height: '500px',
      imagePercent:  100,
      thumbnailsColumns: 4,
      imageAnimation: NgxGalleryAnimation.Slide,
      preview: false
    }
  ]
}

getImages(): NgxGalleryImage[] {
  const imageUrls = [];
  for (const photo of this.product.photos) {
    imageUrls.push({
      small: photo?.url,
      medium: photo?.url,
      big: photo?.url
    })
  }
  return imageUrls;
}

loadProduct() {
  this.productService.getProductById(+this.route.snapshot.paramMap.get('id')).subscribe(product => {
    this.product = product;
    localStorage.setItem("product",JSON.stringify(product));
    this.galleryImages = this.getImages();
  }, error => {console.log(error); })
}

deleteProduct() {
  return this.productService.deleteProduct(this.product.id).subscribe(response => {
    console.log(response);
  }, error => {console.log(error);} );
}
```

I also created a porducts service that is responsible of sending http requests to the back-end.

Sayfa No	Çalışmanın		KONTROL
16	Konusu: Created product details component.	Yapıldığı Tarih 22/03/2022	.... Kaan KANDEMİR Bilgi İşlem

Today I added the functionality of adding photos to products in the front-end by creating a new component then I used the Ng2FileUploader that angular provides. Even though we can use it out of the box it was really difficult to configure it and it took me very long.

The FileUploader component:

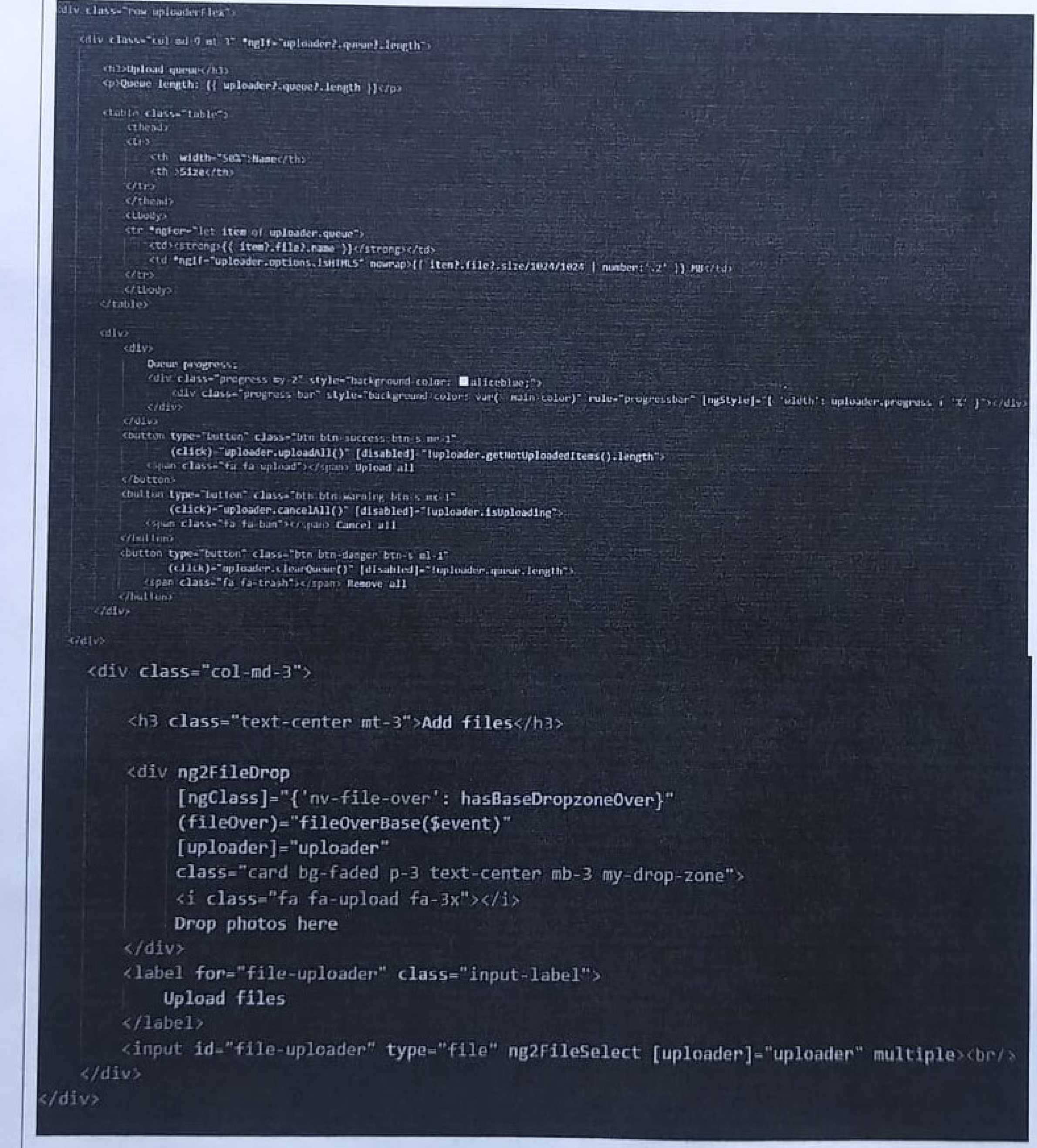


```
export class PhotoEditorComponent implements OnInit {
  baseUrl = environment.apiUrl;
  @Input() product: Product;
  photo: Photo;
  params = new HttpParams();
  uploader: FileUploader;
  hasBaseDropzoneOver = false;

  constructor(private productService: ProductsService) { }

  ngOnInit(): void {
    this.initializeUploader();
  }

  setParams() {
    this.params = this.params.set("productId", this.product.id)
    this.params = this.params.set("photoId", this.photo.id);
  }

  deletePhoto(photo:Photo) {
    this.photo = photo;
    this.setParams();
    this.productService.deletePhoto(this.params).subscribe(() => {
      this.product.photos.slice[photo.id];
    })
  }

  setMainPhoto(photo:Photo) {
    this.photo = photo;
    this.setParams();
    this.productService.updateMainPhoto(this.params).subscribe(response => {
      console.log(response);
    })
  }

  fileOverBase(e: any) {
    this.hasBaseDropzoneOver = e;
  }
  initializeUploader() {
    this.uploader = new FileUploader({
      url: this.baseUrl + "products/add-photo?id=" + this.product.id,
      // authToken: 'Bearer ' + this.user.token, // after adding users
      isHTML5: true,
      allowedFileType: ['image'],
      removeAfterUpload: true,
      autoUpload: false,
      maxFileSize: 10 * 1024 * 1024,
    });

    this.uploader.onAfterAddingFile = (file) => {
      file.withCredentials = false;
    }


    this.uploader.onSuccessItem = (item, response, status, headers) => {
      if(response) {
        const photo = JSON.parse(response);
        this.product.photos.push(photo);
      }
    }
  }
```

Sayfa No	Çalışmanın		KONTROL
17	Konusu: Created photo uploader.	Yapıldığı Tarih 23/03/2022	.... Kaan KANDEMİR Bilgi İşlem

Today I completed my photo uploader by adding the HTML codes needed and passing all the values between the html index and it's Typescript component properly.

The HTML index:

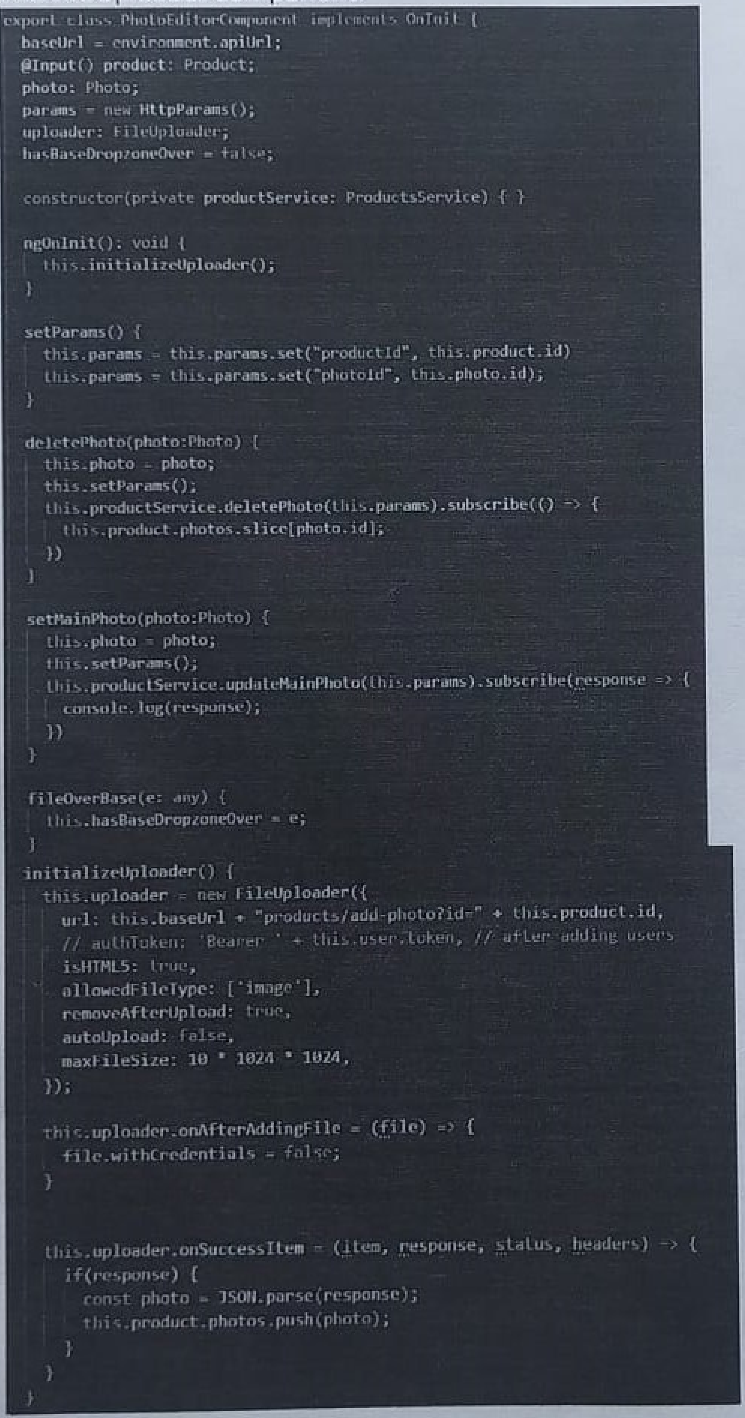


```
<div class="row uploaderflex">

    <div class="col-md-9 mt-3" *ngIf="uploader?.queue?.length">

        <h3>Upload queue</h3>
        <p>Queue length: {{ uploader?.queue?.length }}</p>

        <table class="table">
            <thead>
            <tr>
                <th width="50%">Name</th>
                <th >Size</th>
            </tr>
            </thead>
            <tbody>
            <tr *ngFor="let item of uploader.queue">
                <td><strong>{{ item?.file?.name }}</strong></td>
                <td *ngIf="uploader.options.isHTML5" nowrap>{{ item?.file?.size/1024/1024 | number:'.2' }} MB</td>
            </tr>
            </tbody>
        </table>

        <div>
            <div>
                Queue progress:
                <div class="progress my-2" style="background-color: ■aliceblue;">
                    <div class="progress-bar" style="background-color: var(--main-color)" role="progressbar" [ngStyle]="{ 'width': uploader.progress + '%' }"></div>
                </div>
            </div>
            <button type="button" class="btn btn-success btn-s me-1"
                    (click)="uploader.uploadAll()" [disabled]="!uploader.getNotUploadedItems().length">
                <span class="fa fa-upload"></span> Upload all
            </button>
            <button type="button" class="btn btn-warning btn-s me-1"
                    (click)="uploader.cancelAll()" [disabled]="!uploader.isUploading">
                <span class="fa fa-ban"></span> Cancel all
            </button>
            <button type="button" class="btn btn-danger btn-s ml-1"
                    (click)="uploader.clearQueue()" [disabled]="!uploader.queue.length">
                <span class="fa fa-trash"></span> Remove all
            </button>
        </div>

    </div>
    <div class="col-md-3">

        <h3 class="text-center mt-3">Add files</h3>

        <div ng2FileDrop
             [ngClass]="{'nv-file-over': hasBaseDropzoneOver}"
             (fileOver)="fileOverBase($event)"
             [uploader]="uploader"
             class="card bg-faded p-3 text-center mb-3 my-drop-zone">
             <i class="fa fa-upload fa-3x"></i>
             Drop photos here
        </div>
        <label for="file-uploader" class="input-label">
            Upload files
        </label>
        <input id="file-uploader" type="file" ng2FileSelect [uploader]="uploader" multiple><br/>
    </div>
</div>
```

Sayfa No	Çalışmanın		KONTROL
18	Konusu: Making the HTML part of the photo uploader.	Yapıldığı Tarih 24/03/2022	.... Kaan KANDEMİR Bilgi İşlem

Today I added the functionality of adding photos to products in the front-end by creating a new component then I used the Ng2FileUploader that angular provides. Even though we can use it out of the box it was really difficult to configure it and it took me very long.

The FileUploader component:

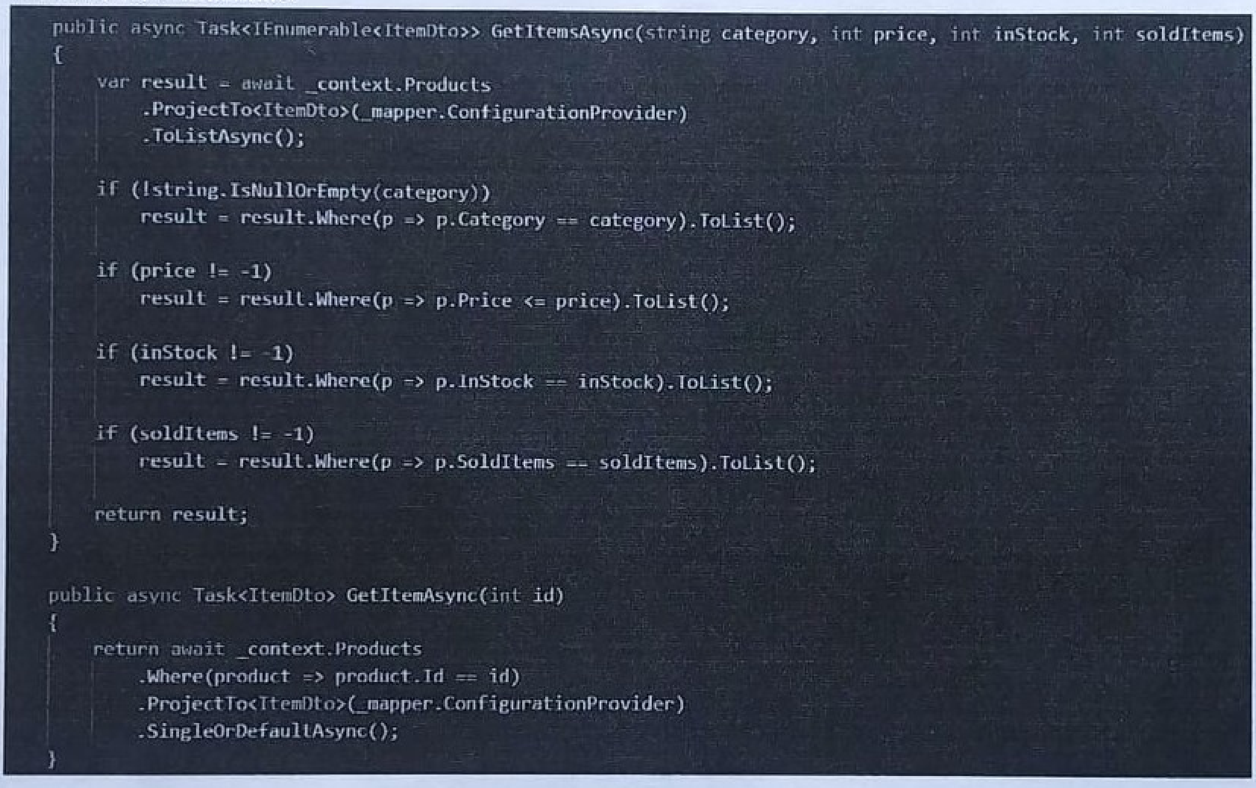


```
export class PhotoEditorComponent implements OnInit {
  baseUrl = environment.apiUrl;
  @Input() product: Product;
  photo: Photo;
  params = new HttpParams();
  uploader: FileUploader;
  hasBaseDropzoneOver = false;

  constructor(private productService: ProductsService) { }

  ngOnInit(): void {
    this.initializeUploader();
  }

  setParams() {
    this.params = this.params.set("productId", this.product.id)
    this.params = this.params.set("photoId", this.photo.id);
  }

  deletePhoto(photo:Photo) {
    this.photo = photo;
    this.setParams();
    this.productService.deletePhoto(this.params).subscribe(() => {
      this.product.photos.slice[photo.id];
    })
  }

  setMainPhoto(photo:Photo) {
    this.photo = photo;
    this.setParams();
    this.productService.updateMainPhoto(this.params).subscribe(response => {
      console.log(response);
    })
  }

  fileOverBase(e: any) {
    this.hasBaseDropzoneOver = e;
  }
  initializeUploader() {
    this.uploader = new FileUploader({
      url: this.baseUrl + "products/add-photo?id=" + this.product.id,
      // authToken: 'Bearer ' + this.user.token, // after adding users
      isHTML5: true,
      allowedFileType: ['image'],
      removeAfterUpload: true,
      autoUpload: false,
      maxFileSize: 10 * 1024 * 1024,
    });

    this.uploader.onAfterAddingFile = (file) => {
      file.withCredentials = false;
    }


    this.uploader.onSuccessItem = (item, response, status, headers) => {
      if(response) {
        const photo = JSON.parse(response);
        this.product.photos.push(photo);
      }
    }
}
```

Sayfa No	Çalışmanın		KONTROL
19	Konusu: Adding Product DTO and mapping the Products to it	Tarih: 25/03/2022	Kaan KANDEMIR Bilgi İşlem

Finally after I added the new DTO I had to make new methods in the repository that send products DTOs instead of the products themselves and then made some neccessary modifications on the products controller.

The two new methods:



```
public async Task<IEnumerable<ItemDto>> GetItemsAsync(string category, int price, int inStock, int soldItems)
{
    var result = await _context.Products
        .ProjectTo<ItemDto>(_mapper.ConfigurationProvider)
        .ToListAsync();

    if (!string.IsNullOrEmpty(category))
        result = result.Where(p => p.Category == category).ToList();

    if (price != -1)
        result = result.Where(p => p.Price <= price).ToList();

    if (inStock != -1)
        result = result.Where(p => p.InStock == inStock).ToList();

    if (soldItems != -1)
        result = result.Where(p => p.SoldItems == soldItems).ToList();

    return result;
}

public async Task<ItemDto> GetItemAsync(int id)
{
    return await _context.Products
        .Where(product => product.Id == id)
        .ProjectTo<ItemDto>(_mapper.ConfigurationProvider)
        .SingleOrDefaultAsync();
}
```

Sayfa No	Çalışmanın		KONTROL
20	Konusu: Applying the DTO structure in the API.	Yapıldığı Tarih 28/03/2022	............................../... ....Kaan KANDEMİR Bilgi İşlem