



# STAJ FAALİYET RAPORU

MÜHENDİSLİK FAKÜLTESİ	
STAJ YAPANIN	
Bölümü	: Bilgisayar Mühendisliği
Adı ve Soyadı	: Serkan Çamkerten
Okul Numarası	: 1811012002
Staj Konusu	: Web Geliştirme

**SÜLEYMAN DEMİREL ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**



**Adı ve Soyadı** : Serkan Çamkerten

**Okul Numarası** : 1811012002

**Okuduğu Yarıyıl** : 8

**Staj Konusu** : Yazılım

**Staj Yaptığı Kurum** : Adesso Turkey

Kurumun Adı	Staj Konusu	Çalışma Süresi	
		Tarihinden	Tarihine
Adesso Turkey	Yazılım	02.08.2022	29.08.2022

### **PRATİK ÇALIŞMANIN GÜNLERE GÖRE DAĞITIM CETVELİ**

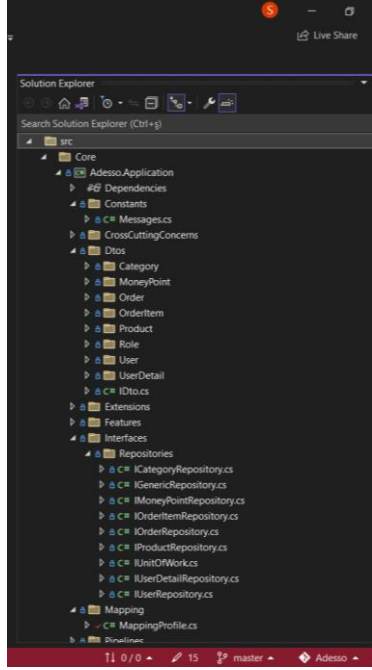
[illegible]

Yapılan İş: Program kurulumu	Sayfa No: 1
	Tarih: 02.08.2022
<p>Tanışma toplantısına katıldım, çalışma arkadaşlarımı tanıdım. Herhangi bir sorun olduğunda kim ile iletişime geçeceğimi, Şirketin projelerin, kullanılan teknolojileri öğrendim. Şirketin vermiş olduğu bilgisayarına gerekli yazılımları ve programları kurdum.</p> <p>Kurulan programlar:</p> <ul style="list-style-type: none"><li>• Visual Studio 2022</li><li>• Visual Studio Code</li><li>• Node.js</li><li>• Git</li><li>• Docker</li><li>• Microsoft SQL Server Management Studio 18</li><li>• MongoDB Compass</li><li>• Postman</li></ul>	
KONTROL:	

Yapılan İş: Veritabanı Tasarımı	Sayfa No: 2
	Tarih: 03.08.2022
<p>Yazılım mimarım ve Buddy'im ile bir toplantı gerçekleştirdik. Toplantıda bana verilecek proje ve kullanılacak teknolojiler konuşuldu. Ana projelere hazır olmam için ana projelerde kullanılan mimari, tasarım desenleri ve yaklaşımları içeren bir proje yapmam istendi. İstedikleri mimariyi kullanarak gerçekleştirmem gereken bir E-Ticaret projesi için veritabanı taslağı hazırladım. Veritabanı taslağı oluşturduktan sonra inceleme gerçekleştirildi ve birkaç değişiklikten sonra o taslak üzerinden devam ettim.</p> <p>Projede kullanacağım teknolojiler:</p> <p>Backend Projesi 1</p> <ul style="list-style-type: none"><li>• C# (.NET 6)</li><li>• MSSQL</li></ul> <p>Backend Projesi 2</p> <ul style="list-style-type: none"><li>• Javascript (Node.js)</li><li>• MongoDB</li></ul> <p>Frontend Projesi 3</p> <ul style="list-style-type: none"><li>• React.Js</li></ul>	
KONTROL:	

Yapılan İş: Core katmanının geliştirilmesi	Sayfa No: 3
	Tarih: 04.08.2022

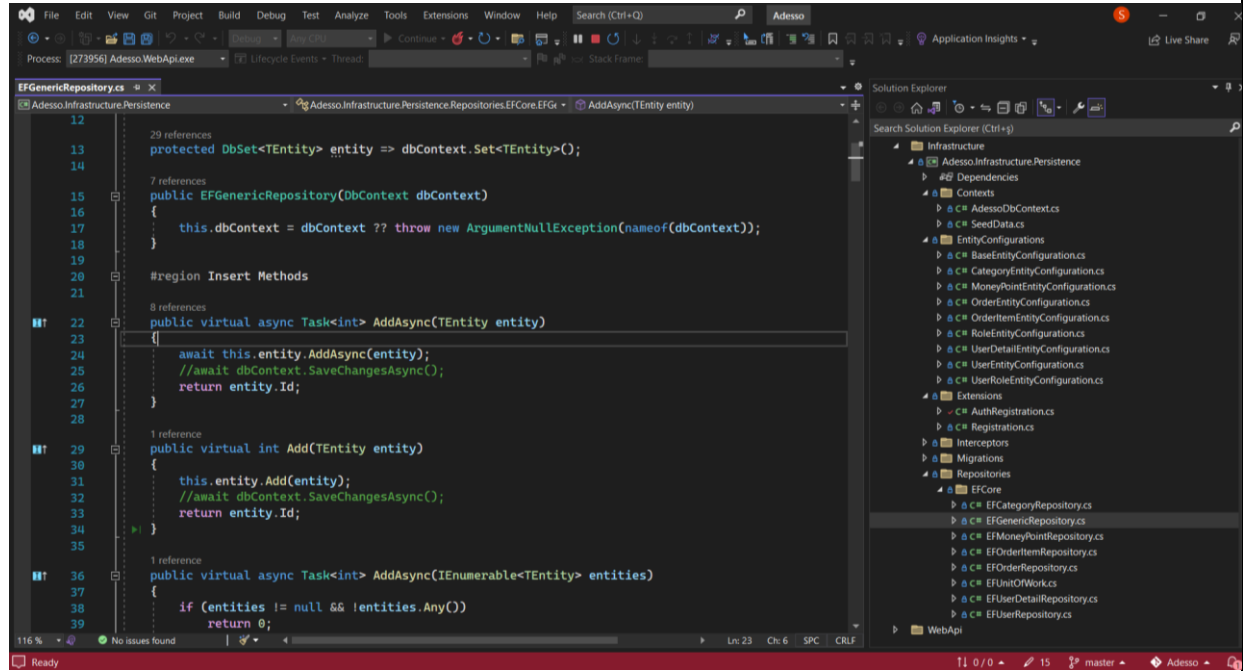
Mimari için gerekli katmanları oluşturdum ve kodlamaya başladım.  
Öncelikli olarak Core katmanını geliştirmeye başladım.  
Gerekli olan Modelleri, DTO'ları, Interface'leri, Mapleme işlemlerini, Extensionları, Constantları yazdım.  
Gün içinde ve ortasında Buddy'im ile bu kodları inceleyerek bazı düzeltmeler ve eklemeler yaptık.



KONTROL:

Yapılan İş: Infrastructure katmanının geliştirilmesi	Sayfa No: 4
	Tarih: 05.08.2022

Mimari içerisindeki Infrastructure katmanını geliştirmeye başladım. Gerekli olan veritabanı-model arasındaki bağlamaları, Context'leri, EntityFramework ile veritabanı işlem depolarını, katman için gerekli olan Extension methodlarını geliştirdim. Her gün olduğu gibi bugün de gün içinde ve ortasında Buddy'im ile bu kodları inceleyerek bazı düzeltmeler ve eklemeler yaptık.



```
12 29 references
13 protected DbSet<TEntity> entity => dbContext.Set<TEntity>();
14
15 7 references
16 public EFGenericRepository(DbContext dbContext)
17 {
18     this.dbContext = dbContext ?? throw new ArgumentNullException(nameof(dbContext));
19 }
20
21 #region Insert Methods
22
23 8 references
24 public virtual async Task<int> AddAsync(TEntity entity)
25 {
26     await this.entity.AddAsync(entity);
27     //await dbContext.SaveChangesAsync();
28     return entity.Id;
29 }
30
31 1 reference
32 public virtual int Add(TEntity entity)
33 {
34     this.entity.Add(entity);
35     //await dbContext.SaveChangesAsync();
36     return entity.Id;
37 }
38
39 1 reference
40 public virtual async Task<int> AddAsync(IEnumerable<TEntity> entities)
41 {
42     if (entities != null && !entities.Any())
43         return 0;
44 }
```

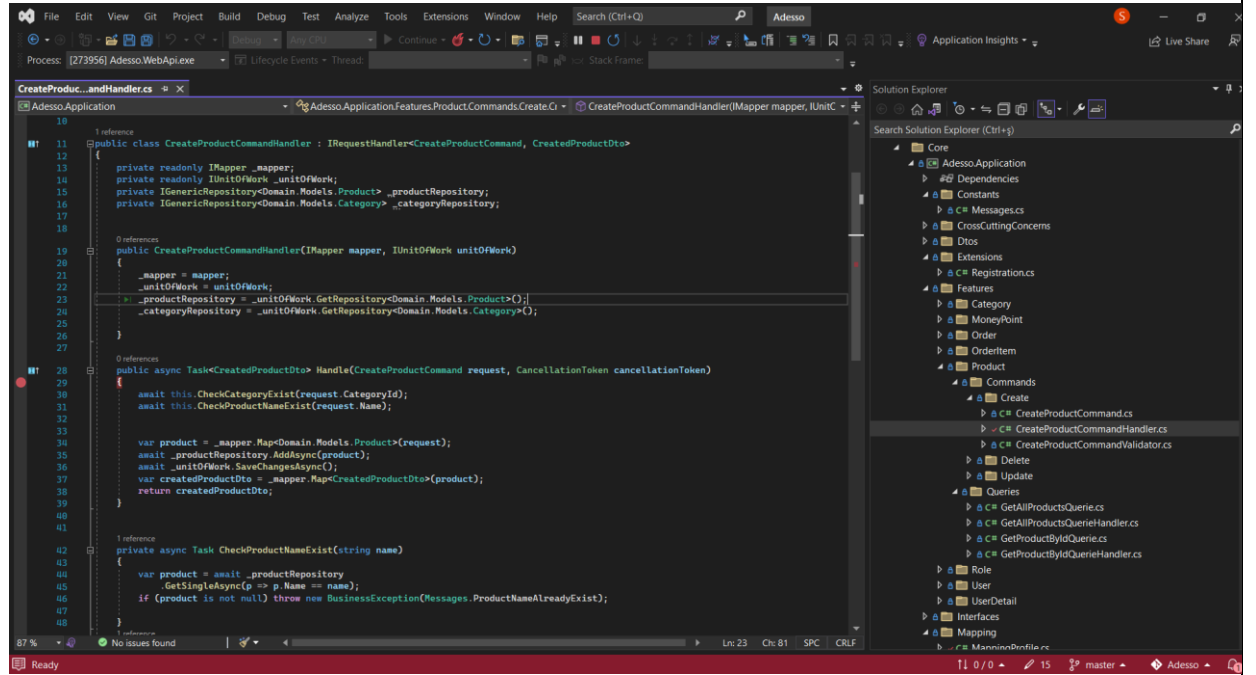
The Solution Explorer on the right shows the project structure:

- Infrastructure
  - Adesso.Infrastructure.Persistence
    - EFCore
      - EFGenericRepository.cs
      - EFMoneyPointRepository.cs
      - EFOrderItemRepository.cs
      - EFUnitOfWork.cs
      - EFUserDetailRepository.cs
      - EFUserRepository.cs

KONTROL:

Yapılan İş: CQRS deseninin uygulanması	Sayfa No: 5
	Tarih: 08.08.2022

Mimari için gerekli API Katmanını oluşturdum ve kodlamaya başladım.  
Bu katmana geçtikten sonra uygulayacak olduğum CQRS deseni için Core katmanına geri döndüm ve Modellerim için gerekli Command ve Querie yapılarını yazdım.  
Daha sonra Kimlik doğrulama, yetkilendirme ve hata yakalama için bazı Exception, Middleware ve Helper methodlar yazdım.



```
11 public class CreateProductCommandHandler : IRequestHandler<CreateProductCommand, CreatedProductDto>
12 {
13     private readonly IMapper _mapper;
14     private readonly IUnitOfWork _unitOfWork;
15     private IGenericRepository<Domain.Models.Product> _productRepository;
16     private IGenericRepository<Domain.Models.Category> _categoryRepository;
17
18     0 references
19     public CreateProductCommandHandler(IMapper mapper, IUnitOfWork unitOfWork)
20     {
21         _mapper = mapper;
22         _unitOfWork = unitOfWork;
23         _productRepository = _unitOfWork.GetRepository<Domain.Models.Product>();
24         _categoryRepository = _unitOfWork.GetRepository<Domain.Models.Category>();
25     }
26
27     0 references
28     public async Task<CreatedProductDto> Handle(CreateProductCommand request, CancellationToken cancellationToken)
29     {
30         await this.CheckCategoryExist(request.CategoryId);
31         await this.CheckProductNameExist(request.Name);
32
33         var product = _mapper.Map<Domain.Models.Product>(request);
34         await _productRepository.AddAsync(product);
35         await _unitOfWork.SaveChangesAsync();
36         var createdProductDto = _mapper.Map<CreatedProductDto>(product);
37         return createdProductDto;
38     }
39
40     1 reference
41     private async Task CheckProductNameExist(string name)
42     {
43         var product = await _productRepository
44             .GetSingleAsync(p => p.Name == name);
45         if (product is not null) throw new BusinessException(Messages.ProductNameAlreadyExist);
46     }
47
48     0 references
```

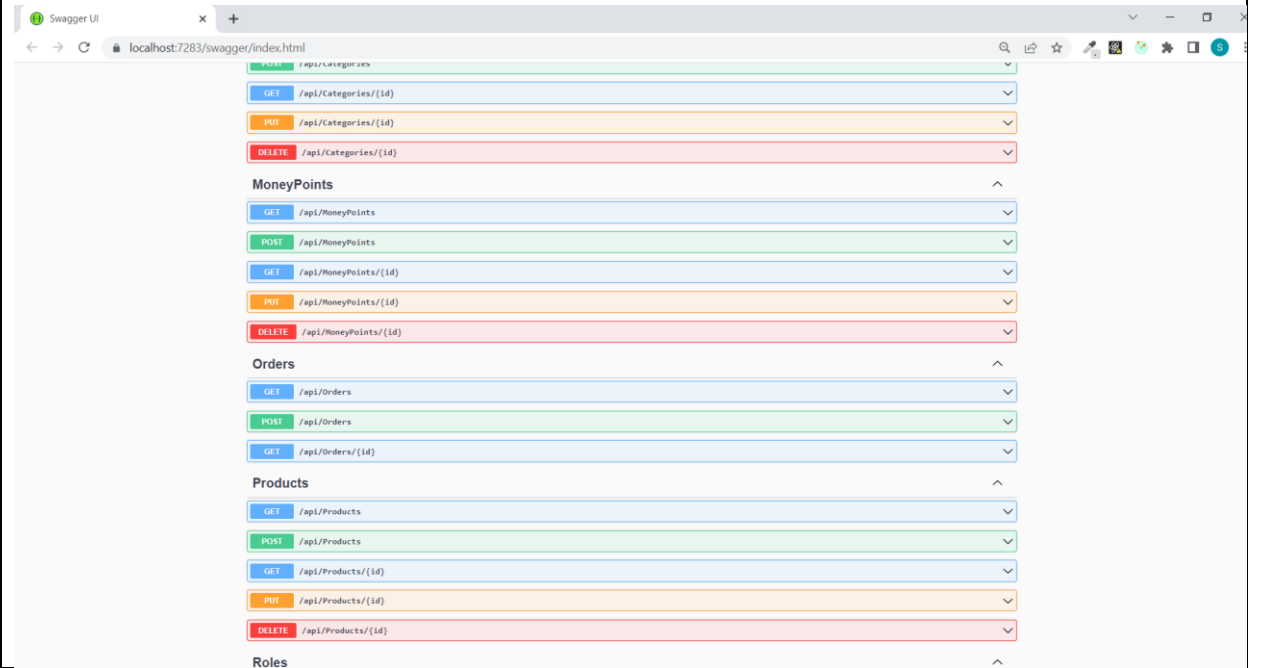
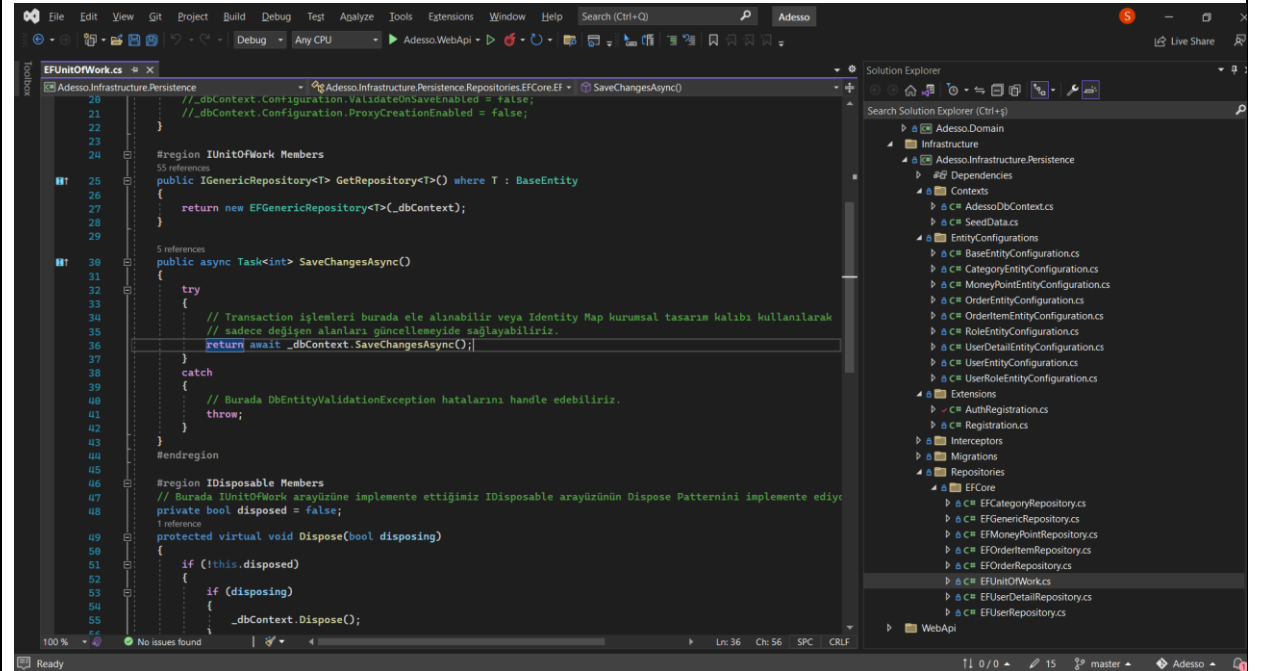
KONTROL:



Yapılan İş: Yazılımın testi	Sayfa No: 6
	Tarih: 09.08.2022

Sistem üzerinde bazı yenilemeler yaptım. Bazı asenkron çalışmayan yapıları asenkron çalışabilecek şekilde düzelttim.

Sistemin testlerini yaptım. Bazı ufak hataları ve isimlendirmelerini değiştirdim. Yeni roller ekledim. Rollerin tanımlamaları için bir Enum yapısı kurdum.



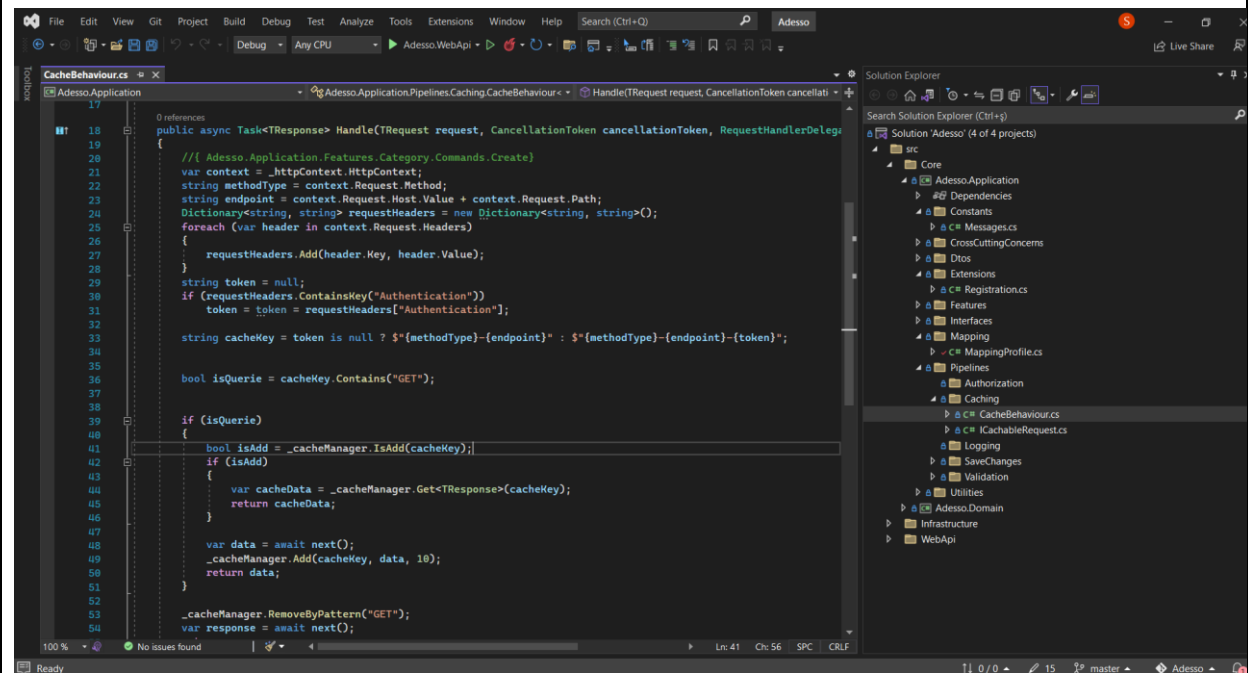
KONTROL:

Yapılan İş: Cross Cutting Concerns(Cache için) ve Pipeline geliştirme	Sayfa No: 7
	Tarih: 10.08.2022

Kullandığımız CQRS deseni içindeki Handler'larda sık kullandığımız (validation, cache, log, save changes, performance vb.) cross cutting concerns'ler gibi yapıları her defasında Handler içinde kullanıp kod tekrarı yapmamak için Mediatr kütüphanesi ile birlikte gelen Handler Pipeline ları araştırdım.

Çok kullanışlı ve gerçekten önemli bir göreve sahip olan bu yapı ile birlikte SaveChanges ve Cache mekanizması için bir algoritma geliştirdim. Bu algoritma öncesinde Cache mekanizmasının sitemimize uygun çalışabilmesi adapter tasarım desenini uyguladım.

Cache için Microsoft ve Redis arasında düşünüp Microsoft'un cache mekanizmasını kendi sistemime göre adapte ettim.



```
0 references
public async Task<Response> Handle(TRequest request, CancellationToken cancellationToken, RequestHandlerDelegate<Response> next)
{
    // Adesso.Application.Features.Category.Commands.Create
    var context = _httpContext.HttpContext;
    string methodType = context.Request.Method;
    string endpoint = context.Request.Host.Value + context.Request.Path;
    Dictionary<string, string> requestHeaders = new Dictionary<string, string>();
    foreach (var header in context.Request.Headers)
    {
        requestHeaders.Add(header.Key, header.Value);
    }
    string token = null;
    if (requestHeaders.ContainsKey("Authentication"))
    {
        token = requestHeaders["Authentication"];
    }

    string cacheKey = token is null ? $"{methodType}-{endpoint}" : $"{methodType}-{endpoint}-{token}";

    bool isQuery = cacheKey.Contains("GET");

    if (isQuery)
    {
        bool isAdd = _cacheManager.IsAdd(cacheKey);
        if (isAdd)
        {
            var cacheData = _cacheManager.Get<Response>(cacheKey);
            return cacheData;
        }

        var data = await next();
        _cacheManager.Add(cacheKey, data, 10);
        return data;
    }

    _cacheManager.RemoveByPattern("GET");
    var response = await next();
}
```

Solution Explorer: Adesso.Application > Pipelines > CacheBehaviour.cs

KONTROL:

Cache datalarının özelleştirilmesi için cache key algoritmasında değişiklikler yaptım.  
Key bilgisi içerisinde request'den gelen ekstra bilgiler bulunduruyorum.  
Artık cache key method tipi, token vb. bilgilerden oluşuyor.  
Sisteme bir loglama sistemi eklendi. Kütüphane kullanmadan bir loglama sistemi yaptım.

The screenshot displays the Visual Studio IDE with two windows open. The top window shows the `LoggingMiddleware.cs` file, which implements a middleware class for logging HTTP requests. The bottom window shows the `log.txt` file, which contains the output of the logging middleware, including the request path, method type, and status code.

```
1 using Adesso.Domain.Enums;
2 using Microsoft.AspNetCore.Http;
3 using Newtonsoft.Json;
4 using System;
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9
10 namespace Adesso.Application.CrossCuttingConcerns.Logging;
11
12 public class LoggingMiddleware
13 {
14     private readonly RequestDelegate _next;
15
16     public LoggingMiddleware(RequestDelegate next)
17     {
18         _next = next;
19     }
20
21     public async Task Invoke(HttpContext context)
22     {
23         context.Response.StatusCode = 200;
24         var logDetailObject = CreateLogDetailHelper.CreateLogDetail(LogLevel.Informational, context, "Başarılı işlemler");
25         string logDetail = JsonConvert.SerializeObject(logDetailObject);
26         Logger.Log(LogTypes.File, logDetail);
27         await _next(context);
28     }
29 }
```

The `log.txt` file contains the following log entries:

```
316 {"LogLevel":3,"Message":"The process cannot access the file 'C:\\Users\\camkerten\\Desktop\\Projects\\AdessoNET\\Adesso'"}
317 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
318 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
319 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
320 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
321 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
322 {"LogLevel":3,"Message":"The process cannot access the file 'C:\\Users\\camkerten\\Desktop\\Projects\\AdessoNET\\Adesso'"}
323 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
324 {"LogLevel":3,"Message":"The process cannot access the file 'C:\\Users\\camkerten\\Desktop\\Projects\\AdessoNET\\Adesso'"}
325 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
326 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
327 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
328 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
329 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
330 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
331 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
332 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
333 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
334 {"LogLevel":3,"Message":"The process cannot access the file 'C:\\Users\\camkerten\\Desktop\\Projects\\AdessoNET\\Adesso'"}
335 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
336 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
337 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
338 {"LogLevel":3,"Message":"The process cannot access the file 'C:\\Users\\camkerten\\Desktop\\Projects\\AdessoNET\\Adesso'"}
339 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
340 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
341 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
342 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
343 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
344 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
345 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
346 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
347 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
348 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
349 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
350 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
351 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
352 {"LogLevel":3,"Message":"The process cannot access the file 'C:\\Users\\camkerten\\Desktop\\Projects\\AdessoNET\\Adesso'"}
353 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
354 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/products/", "MethodType":"GE"}
355 {"LogLevel":6,"Message":"Başarılı işlem","UserId":null,"UserEmailAddress":null,"Path":"/api/categories/", "MethodType":"GE"}
```

KONTROL:

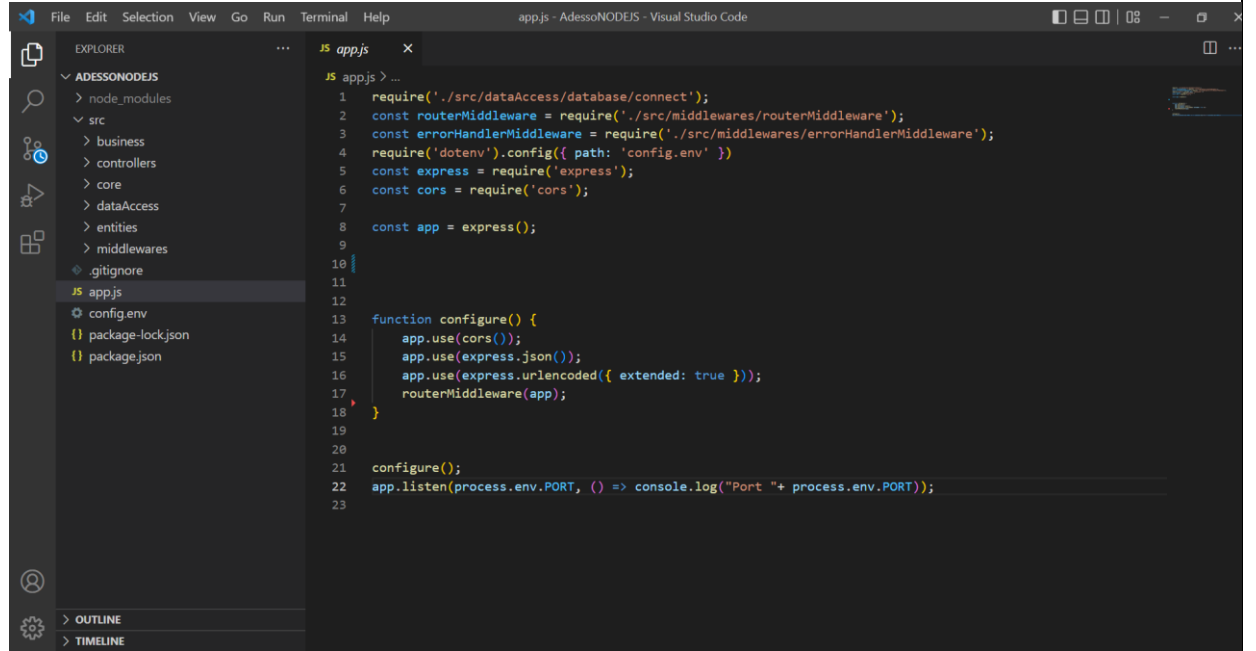
Yapılan İş: Worker service kodlaması	Sayfa No: 9
	Tarih: 12.08.2022
<p>Şirketin ilgilendiği Philip Morris projesine atandım. Takım liderimle tanıştım. Proje ve teknolojiler hakkında konuştuk. Loglama sistemini günlük olarak düzenleyip farklı bir yere alan bir worker service yaptım. Her gün 00.00 'da toplanan loglar o günün tarihini isim alan bir dosyada toplanıp başka bir konuma gönderiliyor.</p>	
KONTROL:	

Yapılan İş: Yeni proje tanımının anlaşılması	Sayfa No: 10
	Tarih: 15.08.2022
<p>Philip Morris Order Management projesi için iş analistleri ve developerlar ile uzun bir toplantı gerçekleştirdik. Projenin genel yapısı ve istenenler anlatıldı.</p> <p>.Net, Node.js ve Vue.js teknolojilerinin kullanılacağı bu sistemde bir çok birbirini tetikleyen ekran bulunmakta. Bu ekranlara ve sistemin geneline hakim olmam için bazı çalışmalar verildi.</p> <p>Sistemin veritabanına daha iyi hakim olabilmek için SQL üzerinde bilgi tazelemesi yaptım.</p> <p>Ayrı olarak gelişimim için ilerlettiğim .NET Backend projemin loglama sistemini editleyerek projeyi bitirdim.</p>	
KONTROL:	

Yapılan İş: Proje anlaşılması ve taslak hazırlama	Sayfa No: 11
	Tarih: 16.08.2022
<p>Philip Morris Order Management projesi için genel taslak bana özel olarak daha ayrıntılı ve tüm yönleriyle anlatıldı.</p> <p>Benim daha ağırlıklı görev alacağım Draft Management ve bir kaç ekran üzerinde daha çok duruldu.</p> <p>Daha önceki sunum kayıtlarını izlemem için paylaşılan dosyayı indirdim ve anlamaya çalıştım.</p> <p>Git reposuna erişim verildi ve inceleme yaptım.</p> <p>Gelişimim için bitirmiş olduğum .NET projesinin aynısını Nodejs-MongoDB tarafında geliştirmek için taslak hazırladım.</p>	
KONTROL:	

Yapılan İş: Proje anlaşılması ve taslak hazırlama	Sayfa No: 12
	Tarih: 17.08.2022

Order Management projesi kapsamında iki tane toplantı gerçekleştirdik.  
Daha çok Admin Panel üzerinde konuşuldu.  
Proje kapsamında proje için kullanacağım ekstra bir bilgisayar daha tanımlandı.  
IT ekibiyle bilgisayar tanımlamalarını yaptık.  
Kendimi geliştirmem için verilen diğer projenin bitirmiş olduğum taslağının kodlamasına başladım.  
NodeJs teknolojisini kullandığım bu projede mimarinin eksileri ve artılarına bakılması için git reposuna pushladım.



```
1 require('./src/dataAccess/database/connect');
2 const routerMiddleware = require('./src/middlewares/routerMiddleware');
3 const errorHandlerMiddleware = require('./src/middlewares/errorHandlerMiddleware');
4 require('dotenv').config({ path: 'config.env' });
5 const express = require('express');
6 const cors = require('cors');
7
8 const app = express();
9
10
11
12
13 function configure() {
14     app.use(cors());
15     app.use(express.json());
16     app.use(express.urlencoded({ extended: true }));
17     routerMiddleware(app);
18 }
19
20
21 configure();
22 app.listen(process.env.PORT, () => console.log("Port " + process.env.PORT));
23
```

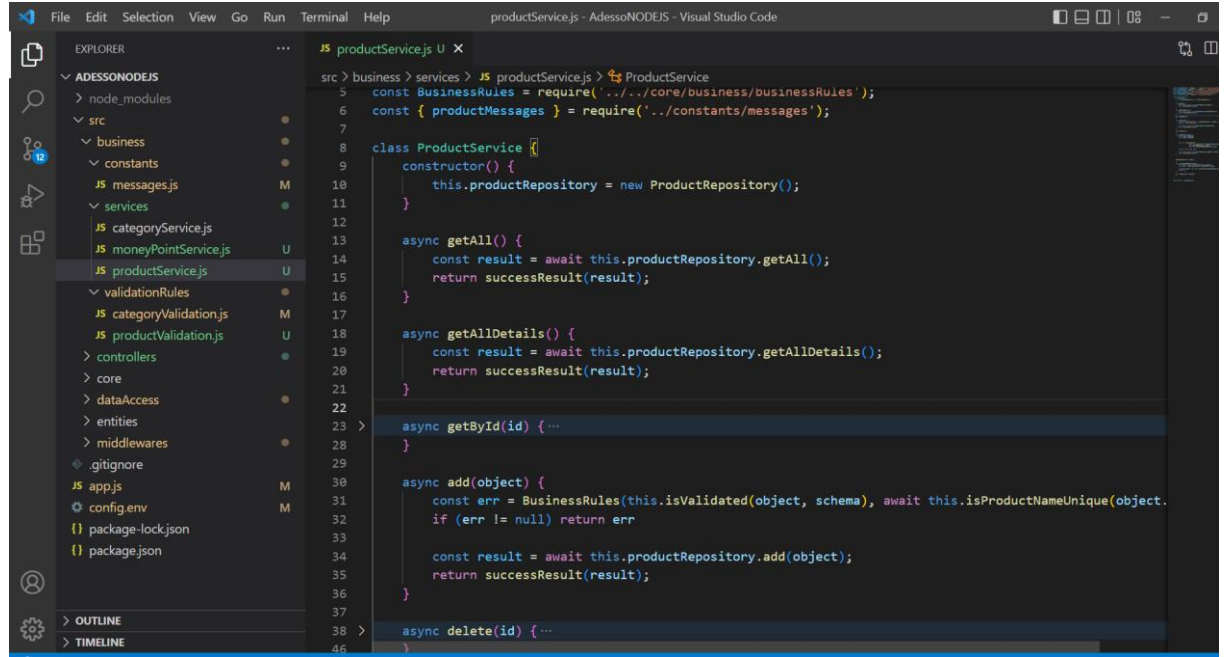
KONTROL:

Yapılan İş: Veritabanı incelemesi, tasarım değişikliği ve kurulumlar	Sayfa No: 13
	Tarih: 18.08.2022
<p>Order Management projesi için proje hakimiyeti açısından veritabanını inceledim. Üstlerim çalışma alanımı, mimariyi, işleyişi ve düzeni anlattı. Günün büyük bir kısmını buralara hakim olmak için çalışmak ile geçti. Ayrı tarafta sürdürdüğüm .NET projesinde mimaride bir kaç değişiklik yapmaya karar verdim ve uygulamak için bir taslak hazırladım. Henüz tam olarak bir değişiklik yapmadım. Nodejs tarafındaki taslakta da değişiklikler yaptım. IT Ekibinin diğer bilgisayara tam erişim sağlamam için gerekli ayarları yapmaları için konuştum ve tam yetki aldım. Proje kapsamında verilen diğer bilgisayara gerekli kurulumları yaptım.</p>	
KONTROL:	



Yapılan İş: Veritabanı incelemesi, tasarım değişikliği ve kurulumlar	Sayfa No: 14
	Tarih: 19.08.2022

BitBucket'dan projeye ait tüm repoları bilgisayarıma çektim.  
Backend, Frontend BFF, EZD. Tüm repoları kabaca inceledim.  
Ağırlıklı olarak çalışacağım alanların üzerinde daha çok durdum.  
Sıklıkla kullanacak olacağımız SQL'de Store Procedure kavramını araştırdım ve daha iyi öğrendim.  
Biraz örnek yaptım.  
Ayrı taraftan geliştirdiğim nodejs projesinin servis katmanını ayarladım.  
Validation kurallarını yazdım ve test ettim.



```
src > business > services > JS productService.js > ProductService
5 const BusinessRules = require('../../core/business/businessRules');
6 const { productMessages } = require('../constants/messages');
7
8 class ProductService {
9   constructor() {
10     this.productRepository = new ProductRepository();
11   }
12
13   async getAll() {
14     const result = await this.productRepository.getAll();
15     return successResult(result);
16   }
17
18   async getAllDetails() {
19     const result = await this.productRepository.getAllDetails();
20     return successResult(result);
21   }
22
23   async getById(id) { ...
24   }
25
26   async add(object) {
27     const err = BusinessRules(this.isValidated(object, schema), await this.isProductNameUnique(object.
28     if (err != null) return err
29
30     const result = await this.productRepository.add(object);
31     return successResult(result);
32   }
33
34   async delete(id) { ...
35   }
36
37
38
39
40
41
42
43
44
45
46
```

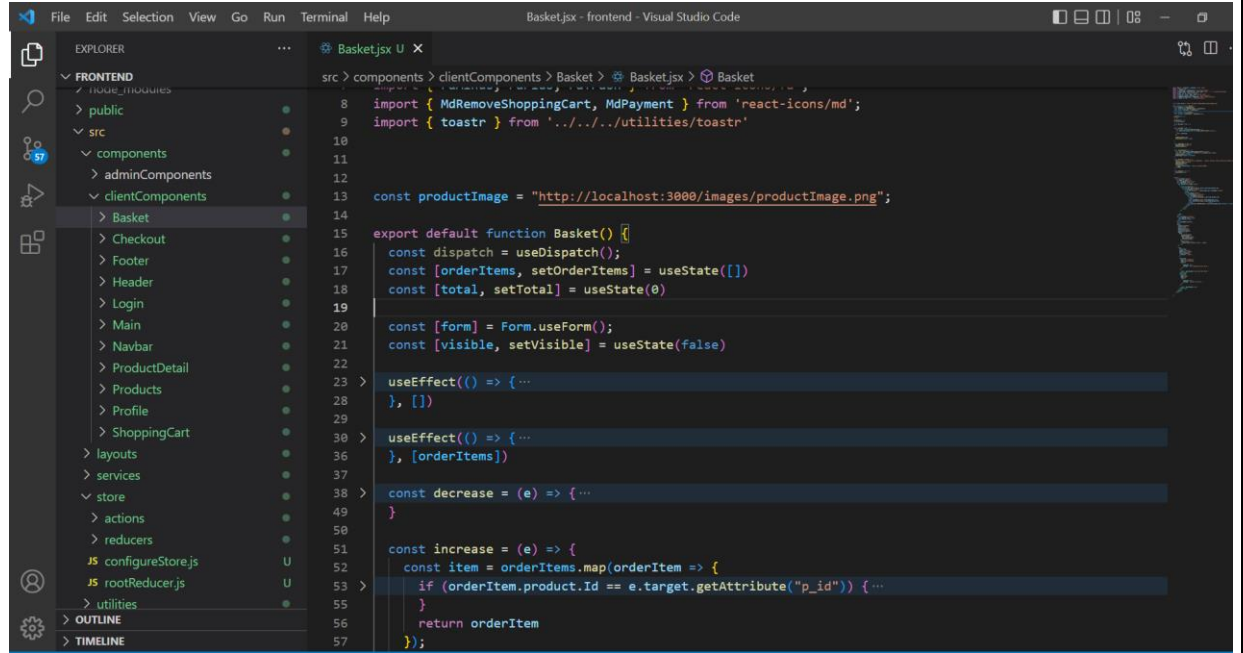
KONTROL:

Yapılan İş: Proje ayaklanması için gerekli ayarlamalar ve araştırma	Sayfa No: 15
	Tarih: 22.08.2022
<p>Repolardaki projeleri ayağa kaldırmak için bir çok kurulum yaptım. NCache, PostSharp .NET Framework versiyonları, IIS Management ayarları, windows features ayarları gibi. Backend, Frontend ve ara katman olan Nodejs ' i ayağa kaldırdım. Yapılan toplantıda bana atılan task'ler için videolar izledim. Eski versiyon ile yapılmış .NET projesini daha derinden anlamaya çalıştım. Kendi gelişimim için verilen ayrı proje için daha gelişmiş sayfalama vb. gibi özelliklere sahip bir Repository yapısı inceledim.</p>	
KONTROL:	

Yapılan İş: Servis güncelleme ve Dosya değişikliğini otomatik yapması gereken uygulama yapımı	Sayfa No: 16
	Tarih: 23.08.2022
<p>Proje için verilen bilgisayarda nCache için her açılıp kapandığında ipconfig'den yenilenen IP'yi alıp, 2 tane dosyada bir çok değişiklik yapılması ve windows servislerinin güncellenmesi gerekiyordu.</p> <p>Her işlemi yönetici olarak çalıştır deyip email ve parola girilince iş çok uzuyordu.</p> <p>Bir uygulama yaptım. ipconfig.exe ' yi okuyup istediğim IP yi alıp gerekli bütün diğer dosyalardaki eski IP ile yer değiştiriyor.</p> <p>Windows servislerini güncelliyor.</p> <p>Ayrı yürüttüğüm .NET projesinde Gelişmiş bir Exception mekanizması entegre ettim.</p> <p>Atanmış olduğum projenin mimarisi ve işleyişi hakkında daha çok şey öğrendim.</p>	
KONTROL:	

Yapılan İş: React.js ile ara yüz kodlaması	Sayfa No: 17
	Tarih: 24.08.2022

Ayrı yürütmem için verilen projenin önyüzünü yazmaya başladım.  
React.js teknolojisi ile bir çok ekranın kodlamasını bitirdim.  
State yönetimi için Redux entegrasi yaptım.  
Jira'a üzerinden bana atanan görevleri aldım ve onlar üstünde çalışmalar gerçekleştirdim.  
Projemizde bulunan veritabanı rowları için yapılmış olan mapleme işlemlerine hakim olabilmek için çalışma yaptım.  
Proje üzerinde debug işlemleri gerçekleştirdim.



The screenshot shows a Visual Studio Code editor with a project named 'Basket.jsx - frontend'. The Explorer panel on the left shows the project structure, including 'FRONTEND', 'public', 'src', 'components', 'adminComponents', 'clientComponents', 'Basket', 'Checkout', 'Footer', 'Header', 'Login', 'Main', 'Navbar', 'ProductDetail', 'Products', 'Profile', 'ShoppingCart', 'layouts', 'services', 'store', 'actions', 'reducers', 'configureStore.js', 'rootReducer.js', 'utilities', 'OUTLINE', and 'TIMELINE'. The main editor area shows the code for the 'Basket' component, which includes imports for 'MdRemoveShoppingCart', 'MdPayment', and 'toastr', a constant for 'productImage', and a default function 'Basket()' that uses 'useState' and 'useEffect' to manage state and side effects.

```
src > components > clientComponents > Basket > Basket.jsx
import { MdRemoveShoppingCart, MdPayment } from 'react-icons/md';
import { toastr } from '../../utilities/toastr'

const productImage = "http://localhost:3000/images/productImage.png";

export default function Basket() {
  const dispatch = useDispatch();
  const [orderItems, setOrderItems] = useState([])
  const [total, setTotal] = useState(0)

  const [form] = Form.useForm();
  const [visible, setVisible] = useState(false)

  useEffect(() => { ...
}, [])

  useEffect(() => { ...
}, [orderItems])

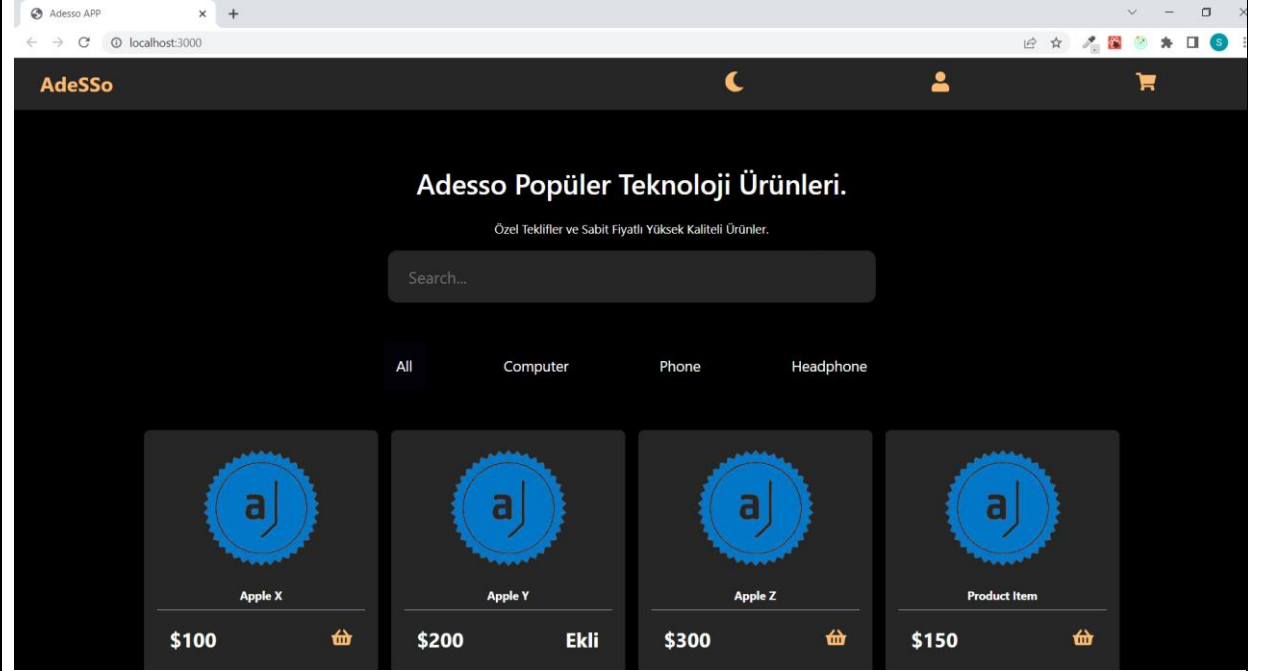
  const decrease = (e) => { ...
}

  const increase = (e) => {
    const item = orderItems.map(orderItem => {
      if (orderItem.product.Id == e.target.getAttribute("p_id")) { ...
    }
  }
  return orderItem
});
```

KONTROL:

Yapılan İş: Ön yüz kodlaması	Sayfa No: 18
	Tarih: 25.08.2022

Atandığım projeye hakim olabilmek için içerdiği diğer projelere debug yaptım.  
İşleyişi öğrenmeye çalıştım.  
Projenin barındırdığı 120 yakın proje var. (.NET solution içerisinde oluşturulan)  
Veri erişim katmanı ve İş katmanı arasındaki bağlantıyı ve işleyişi kavramaya başladım.  
Kendimi geliştirmek için verilmiş olan ve ayrı yürüttüğüm projenin ön yüzüne kısmına başlamıştım. Sipariş ekranının backend'in de değişiklikler yaptım ve redux'ı dahil ederek ön yüz kodlarını yazdım.



KONTROL:

Yapılan İş: Mimari değişikliği ve NetSqlAzman yönetimi	Sayfa No: 19
	Tarih: 26.08.2022
<p>Express.js ile yapmış olduğum rest api uygulamasında mimari değişikliği yaptım. Router-Controller arasındaki bağlantıyı daha uygun bir ve olması gereken duruma getirdim. Veri erişim katmanımı düzenledim. Servis yapısını düzenledim. Bulunmuş olduğum proje kapsamında rol tabanlı yetkilendirmeye hakim olmak için NetSqlAzman yapısını inceledim. Uygulamalı olarak bana gösterildi.</p>	
KONTROL:	

Yapılan İş: Veritabanı yapısı incelemsi	Sayfa No: 20
	Tarih: 29.08.2022
<p>Çalışmış olduğum projenin veritabanı yapısını tekrar inceledim. Tablo isimlendirmeleri, yabancı anahtar isimlendirmeleri sütun ve özellik isimlendirmeleri, tablolarda ortak kullanılan özellikler ve neden kullanıldıkları, diğer veritabanları ile arasındaki ilişkiler, yapılan değişiklikten diğer veritabanının etkilenmesi gibi bir çok konunun üzerinde durarak projenin veritabanları üzerinde işlem yapacak düzeyde bilgi sahibi olmaya çalıştım. Proje ekranları üzerinden tekrar debug yaparak işleyişi daha net anlamaya çalıştım. Ayrı olarak yürüttüğüm .NET, Nodejs ve React projelerinin son dokunuşlarını yapıp sunuma hazır hale getirdim.</p>	
KONTROL:	