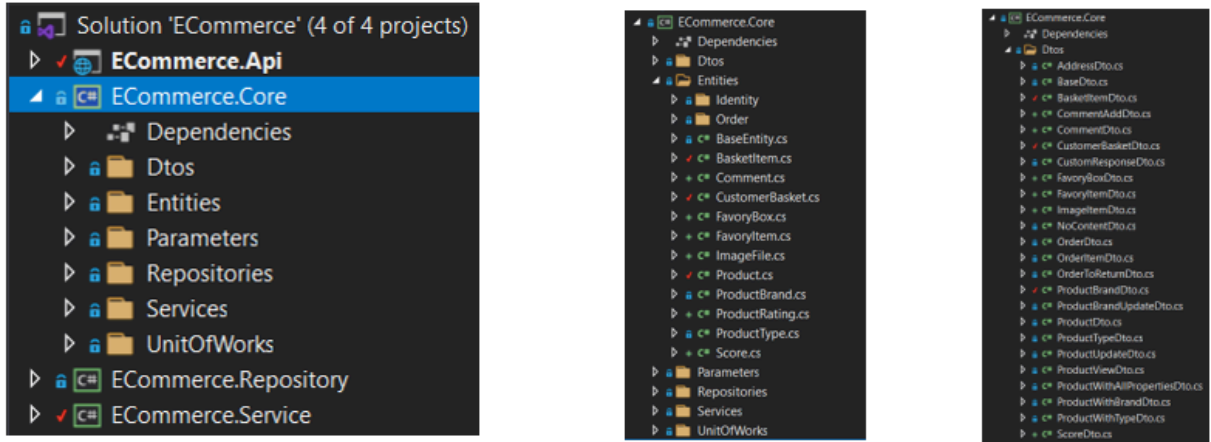
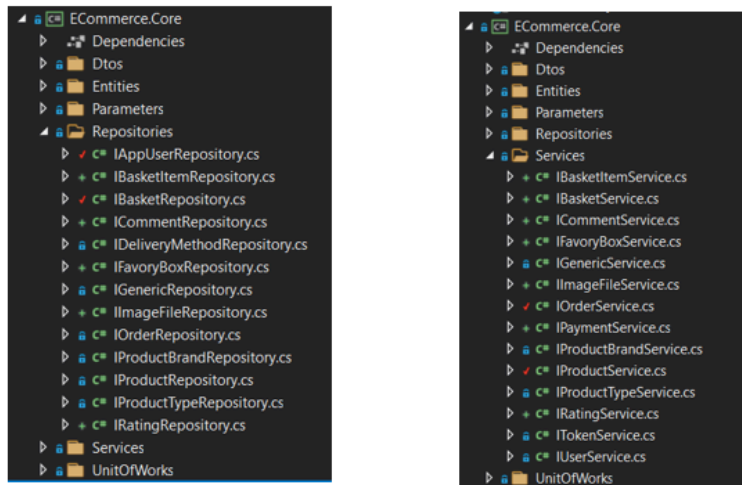


BACKEND **.Net Core API** ve FRONTEND **Angular.js** kullandım

- ❑ C#, .Net Core 3.1 API, Katmanlı mimari, MS Sql Server
- ❑ EntityFrameworkCore, Code First yaklaşımı
- ❑ Dto, Interface, IUnitOfWork ve Repository kullanımı
- ❑ Generic Response Dto nesnesi
- ❑ Email Service
- ❑ Mapping
- ❑ Fluent Validation
- ❑ ANGULAR.JS
- ❑ HTML, CSS, Bootstrap
- ❑ LINQ
- ❑ Identity
- ❑ JWT Token
- ❑ Postman ve Swaagger kullanımı



Şekil 3.4. Ecommerce.Core katmanı görünüşü



Şekil 3.5. Ecommerce.Core layer repository interfaces

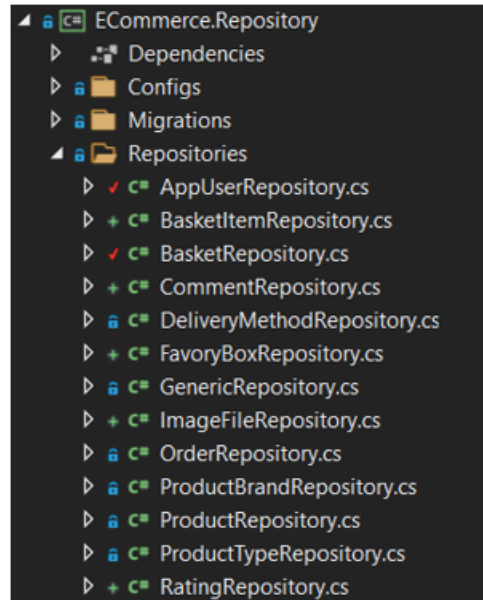
```

public class ECommerceContext : IdentityDbContext<AppUser>
{
    public DbSet<AppUser> AppUsers { get; set; }
    1 reference
    public DbSet<Core.Entities.Identity.Address> Addresses { get; set; }
    9 references
    public DbSet<Product> Products { set; get; }
    0 references
    public DbSet<ProductType> ProductTypes { set; get; }
    0 references
    public DbSet<ProductBrand> ProductBrands { set; get; }
    public DbSet<Order> Orders { set; get; }
    0 references
    public DbSet<OrderItem> OrderItems { set; get; }
    0 references
    public DbSet<DeliveryMethod> DeliveryMethods { set; get; }

    public DbSet<Comment> Comments { get; set; }
    0 references
    public DbSet<ImageFile> ImageFiles { get; set; }
    public DbSet<CustomerBasket> CustomerBaskets { get; set; }
    1 reference
    public DbSet<BasketItem> BasketItem { get; set; }
    public DbSet<FavoryBox> FavoryBoxes { get; set; }
    public DbSet<ProductRating> ProductRatings { get; set; }
    0 references
    public DbSet<Score> Scores { get; set; }
}

```

Şekil 3.6. ECommerceContext sınıfı



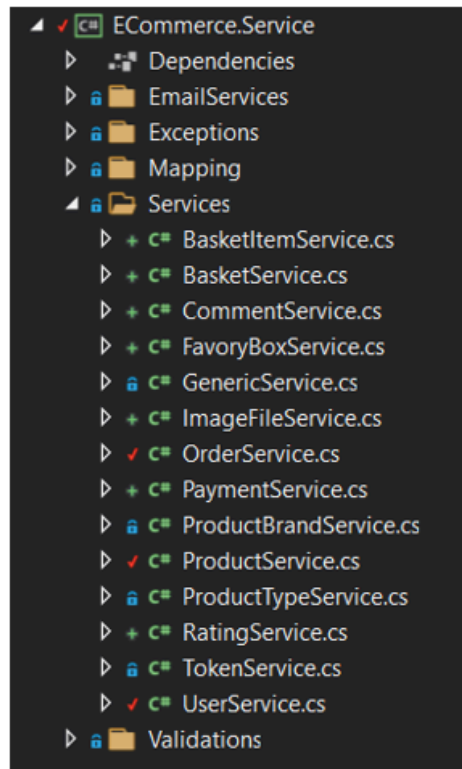
Şekil 3.8. ECommerce.Repository katmanı görünümü

```

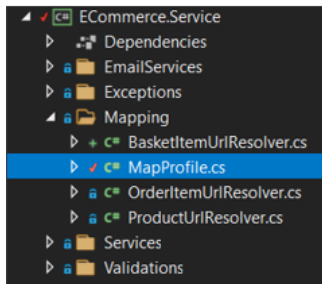
public interface IGenericRepository<T> where T : class
{
    5 references
    IQueryable<T> GetList(Expression<Func<T, bool>> filter = null);
    1 reference
    Task<T> GetAsync(Expression<Func<T, bool>> filter);
    3 references
    Task AddAsync(T entity);
    2 references
    void Delete(T entity);
    2 references
    void Update(T entity);
    7 references
    Task<T> GetByIdAsync(int id);
    2 references
    Task<bool> AnyAsync(Expression<Func<T, bool>> expression);
    2 references
    IQueryable<T> Where(Expression<Func<T, bool>> expression);
    2 references
    void RemoveRange(IEnumerable<T> entities);
    2 references
    Task AddRangeAsync(IEnumerable<T> entities);
}

```

Şekil 3.9. IGenericRepository interface



Şekil 3.10. ECommerce.Service katmanı görünümü



```
public class MapProfile : Profile
{
    0 references
    public MapProfile() {
        var apiUrl = "https://localhost:44347/";

        CreateMap<Product, ProductDto>().ReverseMap();
        CreateMap<ProductUpdateDto, Product>();
        CreateMap<Product, ProductWithBrandDto>();
        CreateMap<Product, ProductWithTypeDto>();
        CreateMap<Product, ProductWithAllPropertiesDto>();

        CreateMap<ProductBrand, ProductBrandDto>().ReverseMap();
        CreateMap<ProductBrandUpdateDto, ProductBrand>();

        CreateMap<ProductType, ProductTypeDto>().ReverseMap();

        CreateMap<Product, ProductViewDto>()
            .ForMember(des => des.ProductBrand, operation => operation.MapFrom(source => source.ProductBrand.Name))
            .ForMember(des => des.ProductType, operation => operation.MapFrom(source => source.ProductType.Name))
            .ForMember(des => des.PictureUrl, operation => operation.ResolveUsing<ProductUrlResolver>());
    }
}
```

Şekil 3.11. MapProfile class

```
0 references
public class RepoServiceModule : Module
{
    0 references
    protected override void Load(ContainerBuilder builder)
    {
        builder.RegisterGeneric(typeof(GenericRepository<>)).As(typeof(IGenericRepository<>)).InstancePerLifetimeScope();
        builder.RegisterGeneric(typeof(GenericService<>)).As(typeof(IGenericService<>)).InstancePerLifetimeScope();

        builder.RegisterType<UnitOfWork>().As<UnitOfWork>();

        var apiAssembly = Assembly.GetExecutingAssembly();
        var repoAssembly = Assembly.GetAssembly(typeof(ECommerceContext)); // o katmandaki herhangi bir dosya ismi verebilirsin o anlar hangi katman olduğunu
        var serviceAssembly = Assembly.GetAssembly(typeof(MapProfile));

        builder.RegisterAssemblyTypes(apiAssembly, repoAssembly, serviceAssembly).Where(x => x.Name.EndsWith("Repository")).AsImplementedInterfaces().InstancePerLifetimeScope();

        builder.RegisterAssemblyTypes(apiAssembly, repoAssembly, serviceAssembly).Where(x => x.Name.EndsWith("Service")).AsImplementedInterfaces().InstancePerLifetimeScope();
    }
}
```

Şekil 3.12. RepoServiceModule class

```
0 reference
public class ProductService : GenericService<Product>, IProductService
{
    private readonly IProductRepository _productRepository;
    private readonly IMapper _mapper;

    0 references
    public ProductService(IGenericRepository<Product> repository, IUnitOfWork unitOfWork, IMapper mapper, IProductRepository productRepository)
    {
        _mapper = mapper;
        _productRepository = productRepository;
    }
}
```

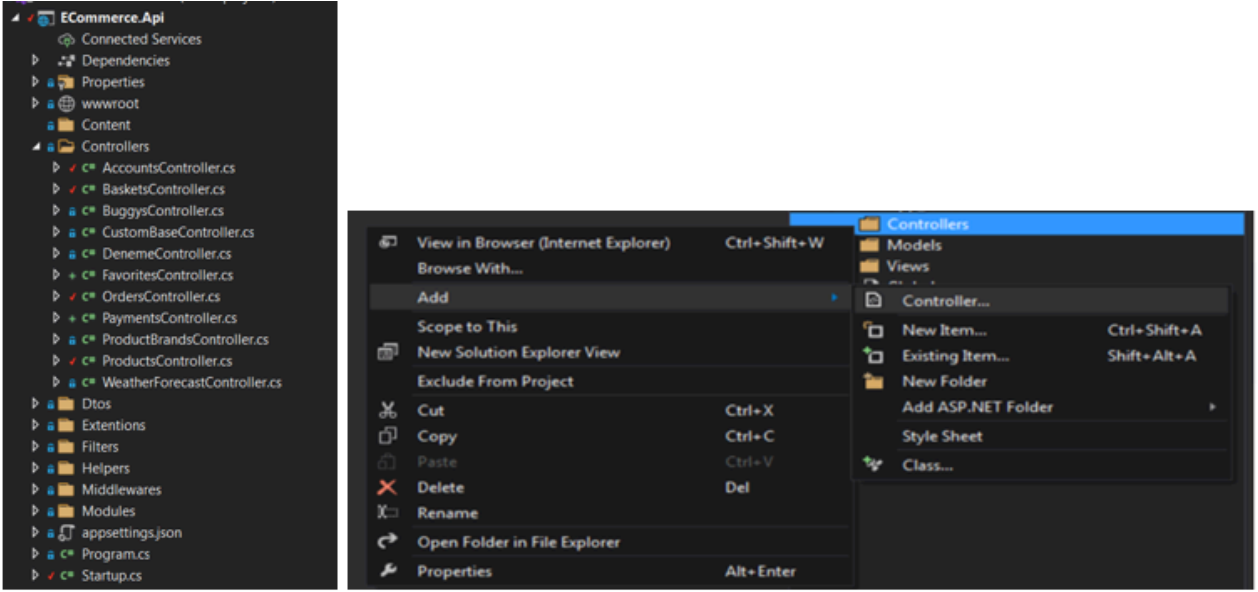
Şekil 3.13. Interface kullanarak dependency injection yapma örneği

```

public class ProductDtoValidator : AbstractValidator<ProductDto>
{
    0 references
    public ProductDtoValidator()
    {
        RuleFor(x => x.Price).InclusiveBetween(1, int.MaxValue).WithMessage("{PropertyName} 0'dan büyük olmalı!").NotEmpty().WithMessage("{PropertyName} 0'dan büyük olmalı!");
        RuleFor(x => x.Stock).InclusiveBetween(1, int.MaxValue).WithMessage("{PropertyName} 0'dan büyük olmalı!");
        RuleFor(x => x.Name).NotNull().WithMessage("{PropertyName} zorunlu alan boş bırakılamaz!").NotEmpty().WithMessage("{PropertyName} zorunlu alan boş bırakılamaz!");
        RuleFor(x->x.ProductBrandId).InclusiveBetween(1, int.MaxValue).WithMessage("{PropertyName} 0'dan büyük olmalı!");
        RuleFor(x->x.ProductTypeId).InclusiveBetween(1, int.MaxValue).WithMessage("{PropertyName} 0'dan büyük olmalı!");
        RuleFor(x => x.Description).NotNull().WithMessage("{PropertyName} zorunlu alan boş bırakılamaz!").NotEmpty().WithMessage("{PropertyName} zorunlu alan boş bırakılamaz!");
    }
}

```

Şekil 3.14. ProductDto için model doğrulama kuralları



Şekil 3.15. Ecommerce.Api katmanında controller oluşturma

```

[Route("api/[controller]")]
[ApiController]
public class ProductsController : CustomBaseController
{
    private readonly IMapper _mapper;
    private readonly IProductService _service;
    private readonly IProductBrandService _productBrandService;
    private readonly IProductTypeService _productTypeService;
    private readonly ICommentService _commentService;
    private readonly IImageFileService _imageFileService;
    private readonly IRatingService _ratingService;

    public ProductsController(IProductService service, IMapper mapper, IProductBrandService productBrandService, IProductTypeService productTypeService,
    {
        _mapper = mapper;
        _service = service;
        _productBrandService = productBrandService;
        _productTypeService = productTypeService;
        _commentService = commentService;
        _imageFileService = imageFileService;
        _ratingService = ratingService;
    }
}

```

```

/// GET api/products
[HttpGet]
public async Task<IActionResult> List()
{
    var products = await _service.GetListAsync();
    var productsDtos = _mapper.Map<List<ProductDto>>(products.ToList());
    return CreateActionResult(CustomResponseDto<List<ProductDto>>.SuccessfulResponse(200, productsDtos));
}

/// GET api/products/2
[HttpGet("{id}")]
public async Task<IActionResult> GetById(int id)
{
    var product = await _service.GetByIdAsync(id);
    var productDto = _mapper.Map<ProductDto>(product);
    return CreateActionResult(CustomResponseDto<ProductDto>.SuccessfulResponse(200, productDto));
}

[HttpPost]
public async Task<IActionResult> Save(ProductDto productDto)
{
    var product = await _service.AddAsync(_mapper.Map<Product>(productDto));
    var dto = _mapper.Map<ProductDto>(product);
    return CreateActionResult(CustomResponseDto<ProductDto>.SuccessfulResponse(201, dto));
}

```

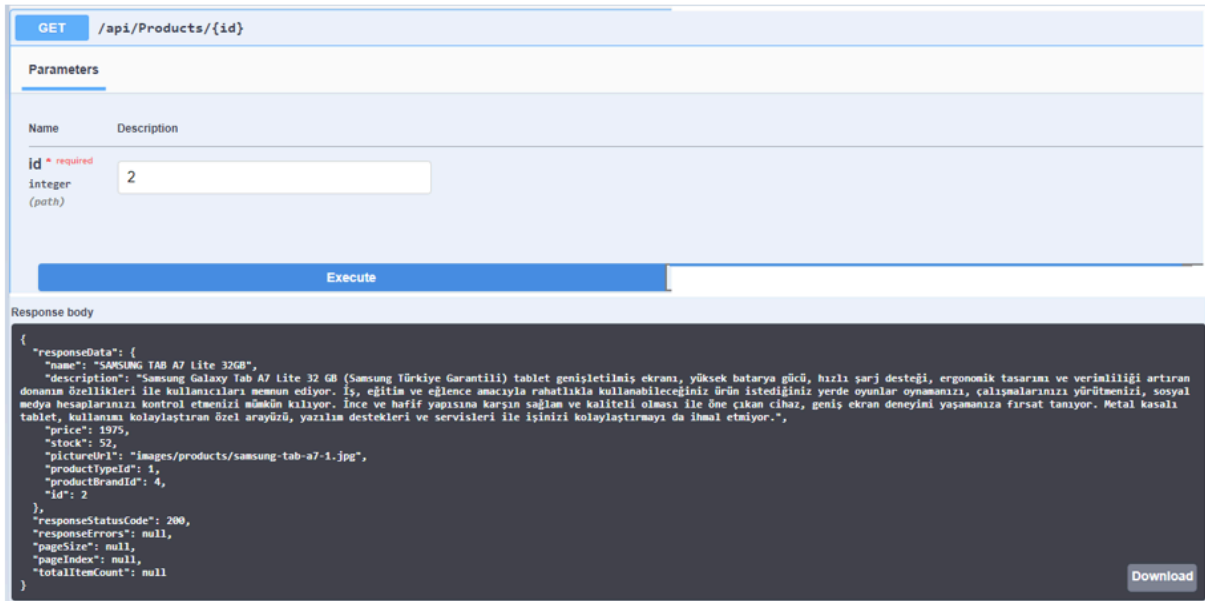
Şekil 3.17. ProductsController görünümü

```

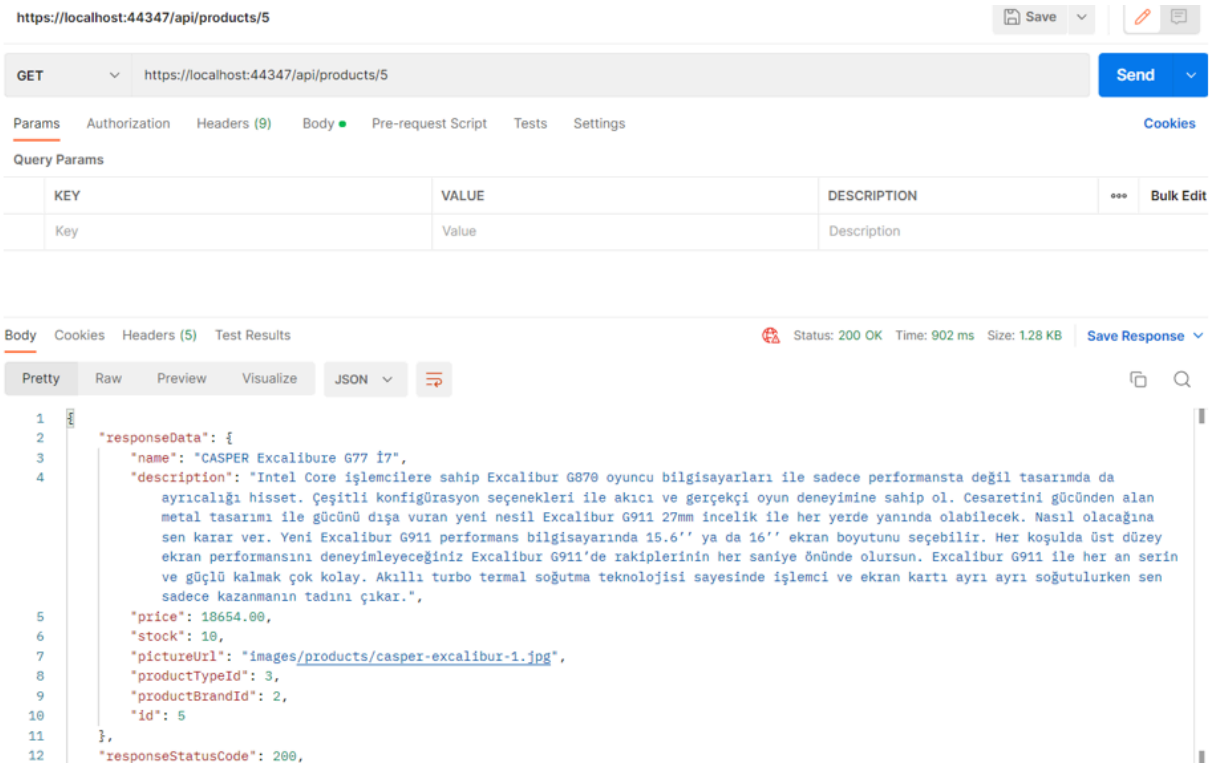
[HttpGet("cheap-products")]
public async Task<IActionResult> CheapProducts()
{
    return CreateActionResult(await _service.GetCheapProducts());
}

```

Şekil 3.18. En ucuz ürünleri getiren action



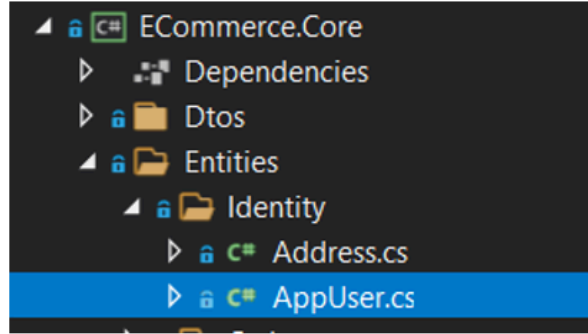
Şekil 3.19. Swagger kullanımı



Şekil 3.20. Postman kullanımı

3.4. Üyelik İşlemleri

ASP.NET Core Identity, ASP.NET Core uygulamalarında kullanıcı hesaplarını yönetmek ve depolamak için bir framework sağlar.

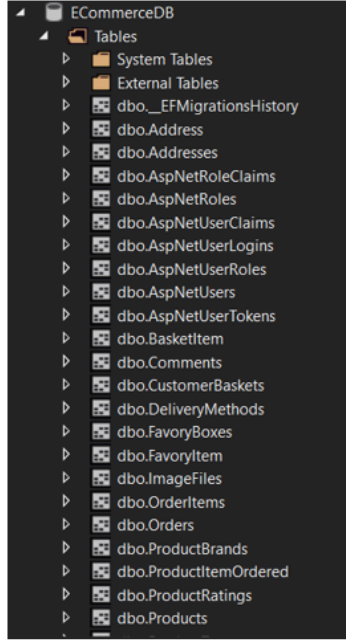


Şekil 3.21. AppUser sınıfı oluşturulması

ECommerce.Core katmanında AppUser class'ı oluşturdum ve IdentityUser interface'inden kalıtım aldım.

```
public class AppUser : IdentityUser
{
    6 references
    public string DisplayName { get; set; }
    4 references
    public Address Address { get; set; }
}
```

Şekil 3.22. AppUser sınıfı görünümü



Şekil 3.24. ECommerceDB veritabanı tablolar

```
public class UserService : GenericService<AppUser>, IUserService
{
    private readonly IAppUserRepository _userRepository;
    private readonly IMapper _mapper;
    private readonly UserManager<AppUser> _userManager;

    0 references
    public UserService(IGenericRepository<AppUser> repository, IUnitOfWork unitOfWork, IMapper mapper, IAppUserRepository userRepository, UserManager<AppUser> userManager)
    {
        _mapper = mapper;
        _userRepository = userRepository;
        _userManager = userManager;
    }

    2 references
    public AppUser GetUserWithAddress(AppUser appUser)
    {
        var user = _userRepository.GetUserWithAddress(appUser);
        return user;
    }

    2 references
    public AppUser UpdateUser(AppUser user, AddressDto dto)
    {
        var newAddress = _mapper.Map<AddressDto, Address>(dto);
        var address = _userRepository.UpdateUserAddress(user, newAddress);
        return user;
    }
}
```

Şekil 3.25. UserService görünümü

```

public class TokenService : ITokenService
{
    private readonly IConfiguration _config; // config deki bilgileri okumak için
    private readonly SymmetricSecurityKey _key; // şifreleme kütüphanesi

    0 references
    public TokenService(IConfiguration config)
    {
        _config = config;
        _key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(_config["Token:Key"]));
    }

    4 references
    public string CreateToken(AppUser user)
    {
        var claim = new List<Claim> // Token oluşturuken kullanıcıya yetkiler vermek için
        {
            new Claim(JwtRegisteredClaimNames.Email,user.Email), // email bilgisiyle eşle
            new Claim(JwtRegisteredClaimNames.GivenName,user.DisplayName), //
        };

        var cred = new SigningCredentials(_key, SecurityAlgorithms.HmacSha512Signature); // key ve şifreleme algoritmasını seçtim
        var tokenDescriptor = new SecurityTokenDescriptor
        {
            Subject = new ClaimsIdentity(claim),
            Expires = DateTime.Now.AddDays(7),
            SigningCredentials = cred,
            Issuer = _config["Token:Issuer"]
        }; // Tanımlamamı yaptım artık oluşturcam

        var tokenHandler = new JwtSecurityTokenHandler();
        var token = tokenHandler.CreateToken(tokenDescriptor);

        return tokenHandler.WriteToken(token);
    }
}

```

Şekil 3.27. Token servis görünümü

```

[HttpPost("register")]
public async Task<ActionResult<UserResponseDto>> Register(RegisterDto registerDto)
{
    if (CheckEmailExistAsync(registerDto.Email).Result.Value)
    {
        return BadRequest( CustomResponseDto<ContentDto>.FailResponse(400, "Bu email adresi zaten kayıtlı!"));
    }

    var user = new AppUser
    {
        DisplayName = registerDto.DisplayName,
        Email = registerDto.Email,
        UserName = registerDto.UserName
    };

    var result = await _userManager.CreateAsync(user, registerDto.Password);
    if (!result.Succeeded)
    {
        return BadRequest(customResponseDto<ContentDto>.FailResponse(401, "HATA! Kullanıcı kayıt etme işlemi gerçekleştirilemedi!"));
    }

    if (result.Succeeded)
    {
        // generate token burada ilk oluşturduk
        var token = await _userManager.GenerateEmailConfirmationTokenAsync(user);
        var uriBuilder = new UriBuilder(_config["ReturnPaths:ConfirmEmail"]);
        var query = HttpUtility.ParseQueryString(uriBuilder.Query);
        query["userId"] = user.Id;
        query["token"] = token;
        uriBuilder.Query = query.ToString();
        var urlString = uriBuilder.ToString();
        var msg = "Lütfen hesabınızı onaylamak için aşağıdaki linke tıklayınız: \n\n" + urlString;
        var senderEmail = _config["ReturnPaths:SenderEmail"];
        await _emailSender.SendEmailAsync(senderEmail, user.Email, "Confirm your email address", msg);

        var favoryBox = new FavoryBox();
        favoryBox.UserId = user.Id;
        var favoryBoxDto = _mapper.Map<FavoryBoxDto>(favoryBox);
        _favoryBoxService.UpdateFavoryBox(favoryBoxDto);

        var basket = new CustomerBasket();
        basket.UserId = user.Id;
        await _basketService.CreateOrUpdateBasketAsync(basket);

        return Ok(new UserResponseDto
        {
            DisplayName = user.DisplayName,
            Email = user.Email,
            Token = _tokenService.CreateToken(user)
        });
    }
}

```



Şekil 3.28. Register(kullanıcı kayıt) işlemi

```

[Route("api/{controller}")]
[ApiController]
public class ProductsController : ControllerBase
{
    private readonly IMapper _mapper;
    private readonly IProductService _service;
    private readonly IProductBrandService _productBrandService;
    private readonly IProductTypeService _productTypeService;
    private readonly ICommentService _commentService;
    private readonly IImageFileService _imageFileService;
    private readonly IRatingService _ratingService;

    // GET api/products
    [HttpGet]
    public async Task<IActionResult> List()
    {
        var products = await _service.GetListAsync();
        var productsDtos = _mapper.Map<List<ProductDto>>(products.ToList());
        return CreateActionResult<CustomResponseDto<List<ProductDto>>>.SuccessfulResponse(200, productsDtos);
    }

    // GET api/products/2
    [HttpGet("{id}")]
    public async Task<IActionResult> GetById(int id)
    {
        var product = await _service.GetByIdAsync(id);
        var productDto = _mapper.Map<ProductDto>(product);
        return CreateActionResult<CustomResponseDto<ProductDto>>.SuccessfulResponse(200, productDto);
    }

    [HttpPost]
    public async Task<IActionResult> Save(ProductDto productDto)
    {
        var product = await _service.AddAsync(_mapper.Map<Product>(productDto));
        var dto = _mapper.Map<ProductDto>(product);
        return CreateActionResult<CustomResponseDto<ProductDto>>.SuccessfulResponse(201, dto);
    }

    [HttpPut]
    public async Task<IActionResult> Update(ProductUpdateDto updateDto)
    {
        await _service.UpdateAsync(_mapper.Map<Product>(updateDto));
        return CreateActionResult<CustomResponseDto<NoContentDto>>.SuccessfulResponse(204);
    }

    //DELETE api/products/2
    [HttpDelete("{id}")]
    public async Task<IActionResult> Delete(int id)
    {
        var product = await _service.GetByIdAsync(id);
        await _service.DeleteAsync(product);
        return CreateActionResult<CustomResponseDto<NoContentDto>>.SuccessfulResponse(204);
    }

    // GET api/products/GetProductsWithBrand
    [HttpGet("{action}")]
    public async Task<IActionResult> GetProductsWithBrand()
    {
        return CreateActionResult(await _service.GetProductsWithBrand());
    }

    // GET api/products/GetProductsWithType
    [HttpGet("{action}")]
    public async Task<IActionResult> GetProductsWithType()
    {
        return CreateActionResult(await _service.GetProductsWithType());
    }
}

```

Şekil 3.30. ProductsController görünümü

```

[Route("api/{controller}")]
[ApiController]
public class BasketsController : ControllerBase
{
    private readonly IBasketService _basketService;
    private readonly IMapper _mapper;
    private readonly UserManager<AppUser> _userManager;

    // GET api/baskets
    [Authorize(AuthenticationSchemes = "Bearer")]
    [HttpGet]
    public async Task<IActionResult> GetBasketById(string id)
    {
        var basket = await _basketService.GetBasketAsync(id);
        if (basket == null)
        {
            return Unauthorized<CustomResponseDto<NoContentDto>>.FailResponse(404, "basket not found");
        }
        return Ok(basket);
    }

    // POST api/baskets
    [Authorize(AuthenticationSchemes = "Bearer")]
    [HttpPost]
    public async Task<IActionResult> UpdateBasket(CustomerBasketDto customerBasket)
    {
        var basket = _mapper.Map<CustomerBasketDto, CustomerBasket>(customerBasket);
        var updatedBasket = await _basketService.CreateOrUpdateBasketAsync(basket);
        return Ok(updatedBasket);
    }

    [Authorize(AuthenticationSchemes = "Bearer")]
    [HttpDelete]
    public async Task DeleteBasket(string basketId)
    {
        await _basketService.DeleteBasketAsync(basketId);
    }

    [Authorize(AuthenticationSchemes = "Bearer")]
    [HttpGet("{action}")]
    public async Task<IActionResult> GetBasketByEmail(string email)
    {
        var user = await _userManager.FindByEmailAsync(email); // database tarafında dogru
        if (user == null)
        {
            return Unauthorized<CustomResponseDto<NoContentDto>>.FailResponse(401, "user not found");
        }

        var basket = await _basketService.GetBasketByUserIdAsync(user.Id);
        var basketDto = _mapper.Map<CustomerBasketDto>(basket);
        return Ok(basketDto);
    }

    [Authorize(AuthenticationSchemes = "Bearer")]
    [HttpPost("{action}")]
    public async Task<IActionResult> AddProductToBasket(AddProductToBasketDto parameters)
    {
        var email = HttpContext.User?.Claims?.FirstOrDefault(x => x.Type == ClaimTypes.Email)?.Value;
        var user = await _userManager.FindByEmailAsync(email); // database tarafında dogru
        var basket = await _basketService.GetBasketByUserIdAsync(user.Id);
        var addedProduct = await _basketService.AddProductToBasketAsync(basket.Id, parameters.productId, parameters.quantity);

        var basketDto = _mapper.Map<CustomerBasketDto>(basket);
        return Ok(basketDto);
    }
}

```

Şekil 3.33. BasketsController görünümü

```
[Authorize(AuthenticationSchemes = "Bearer")]
[Route("api/[controller]")]
[ApiController]
public class OrdersController : ControllerBase
{
    private readonly IOrderService _orderService;
    private readonly IMapper _mapper;

    [HttpPost]
    public async Task<ActionResult> CreateOrder(OrderDto dto)
    {
        var email = HttpContext.User?.Claims?.FirstOrDefault(x => x.Type == ClaimTypes.Email)?.Value;
        var address = _mapper.Map<AddressDto, Address>(dto.ShipToAddress);
        var order = await _orderService.CreateOrderAsync(email, dto.DeliveryMethod, dto.BasketId, address);

        if (order == null)
        {
            return BadRequest(new CustomResponseDto<NoContentDto>.FailResponse(400, "Sipariş oluşturulurken hata oldu!"));
        }

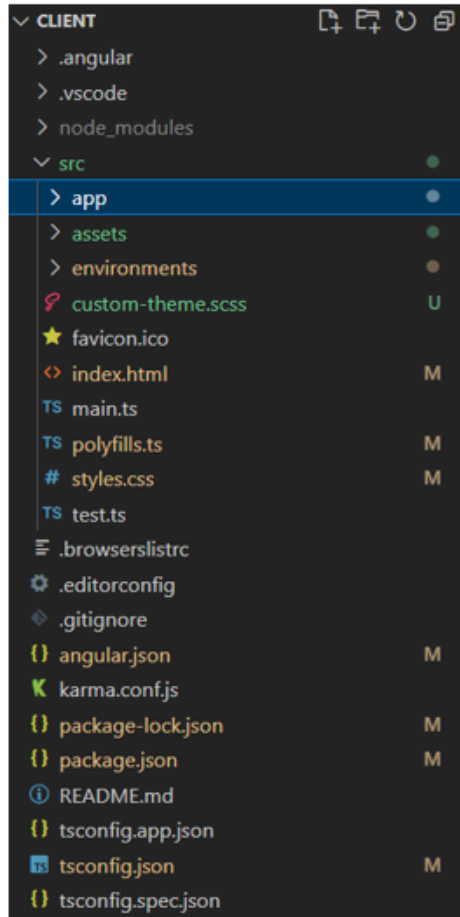
        return Ok(order);
    }

    [HttpGet]
    public async Task<ActionResult> GetOrdersByUser()
    {
        var email = HttpContext.User?.Claims?.FirstOrDefault(x => x.Type == ClaimTypes.Email)?.Value;
        var orders = await _orderService.GetOrdersByUserEmailAsync(email);
        return Ok(_mapper.Map<IReadOnlyList<OrderToReturnDto>>(orders));
    }

    [HttpGet("{id}")]
    public async Task<ActionResult> GetOrderByUserId(int id)
    {
        var email = HttpContext.User?.Claims?.FirstOrDefault(x => x.Type == ClaimTypes.Email)?.Value;
        var order = await _orderService.GetOrderByIdAsync(id, email);
        if (order == null)
        {
            throw new NotFoundException($"{typeof(Order).Name}({id}) order not found");
        }
        return Ok(_mapper.Map<OrderToReturnDto>(order));
    }

    [HttpGet("deliveryMethods")]
    public IActionResult GetDeliveryMethods() // asenkron yapılabilir
    {
        return Ok(_orderService.GetDeliveryMethodAsync());
    }
}
```

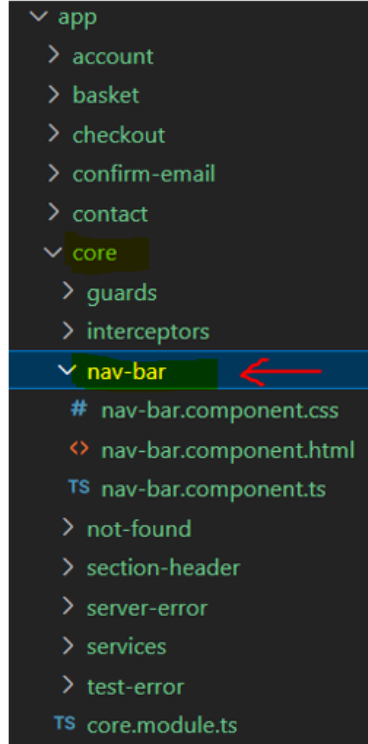
Şekil 3.34. OrdersContoller görünümü



Şekil 5.5. Angular projesi dosya yapısı

```
@NgModule({
  declarations: [
    AppComponent,
    ContactComponent,
    TestComponent
  ],
  imports: [
    BrowserModule,
    //CommonModule,
    AppRoutingModule,
    HttpClientModule,
    CoreModule,
    ShopModule,
    SharedModule,
    BasketModule,
    HomeModule,
    NgxSpinnerModule,
    SwiperModule,
    BrowserAnimationsModule,
    FavoriteModule,
    NgbModule,
    ConfirmEmailModule
  ]
})
```

Şekil 5.6. App.module.ts dosyası görünümü



Şekil 5.7. nav-bar bileşeni (component) oluşturulması

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { ICheapProducts } from '../shared/models/ICheapProducts';
import { IPagination } from '../shared/models/IPagination';

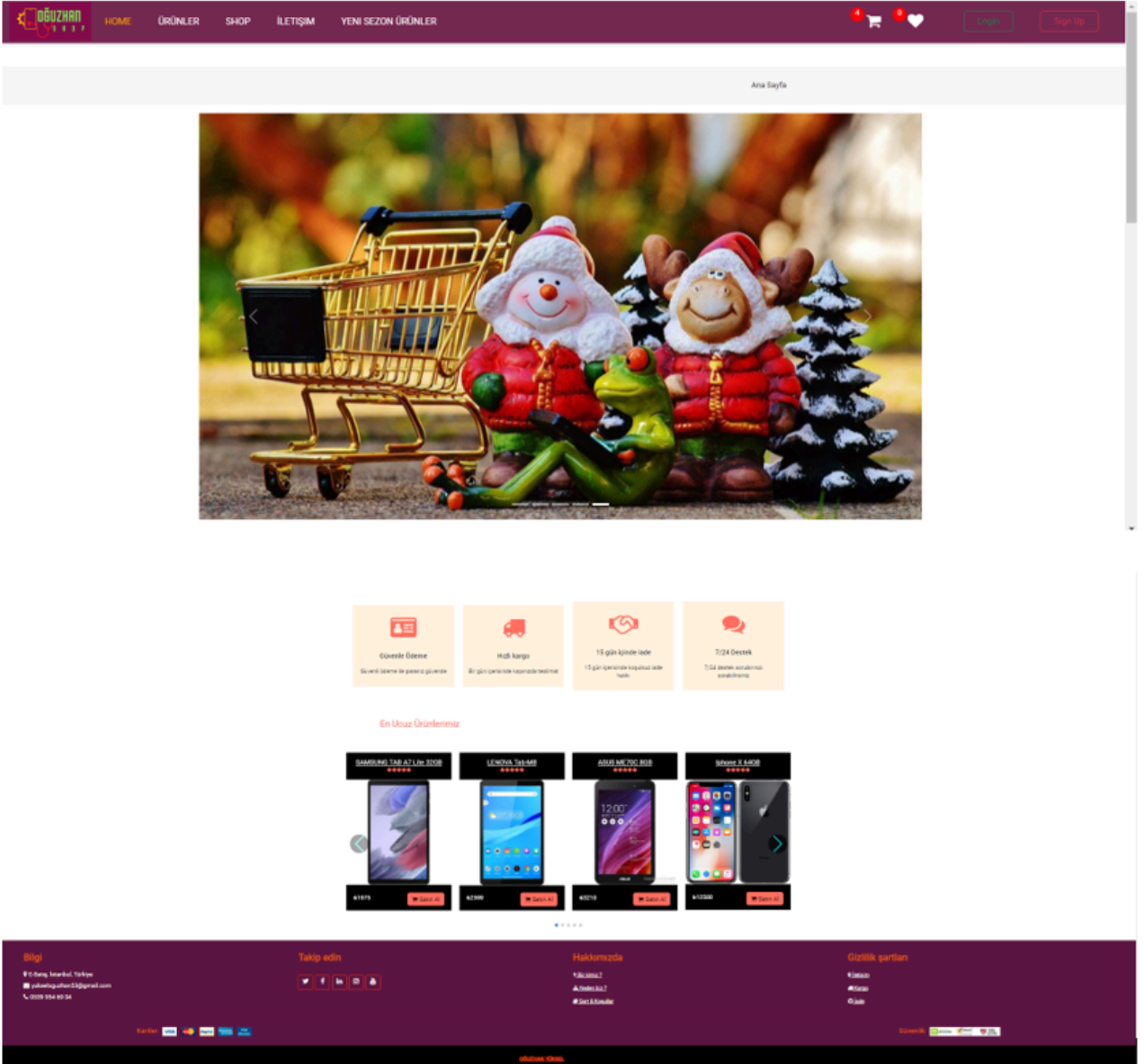
@Injectable({
  providedIn: 'root'
})
export class HomeService {

  baseUrl = "https://localhost:44347/api/";
  constructor( private http : HttpClient) {

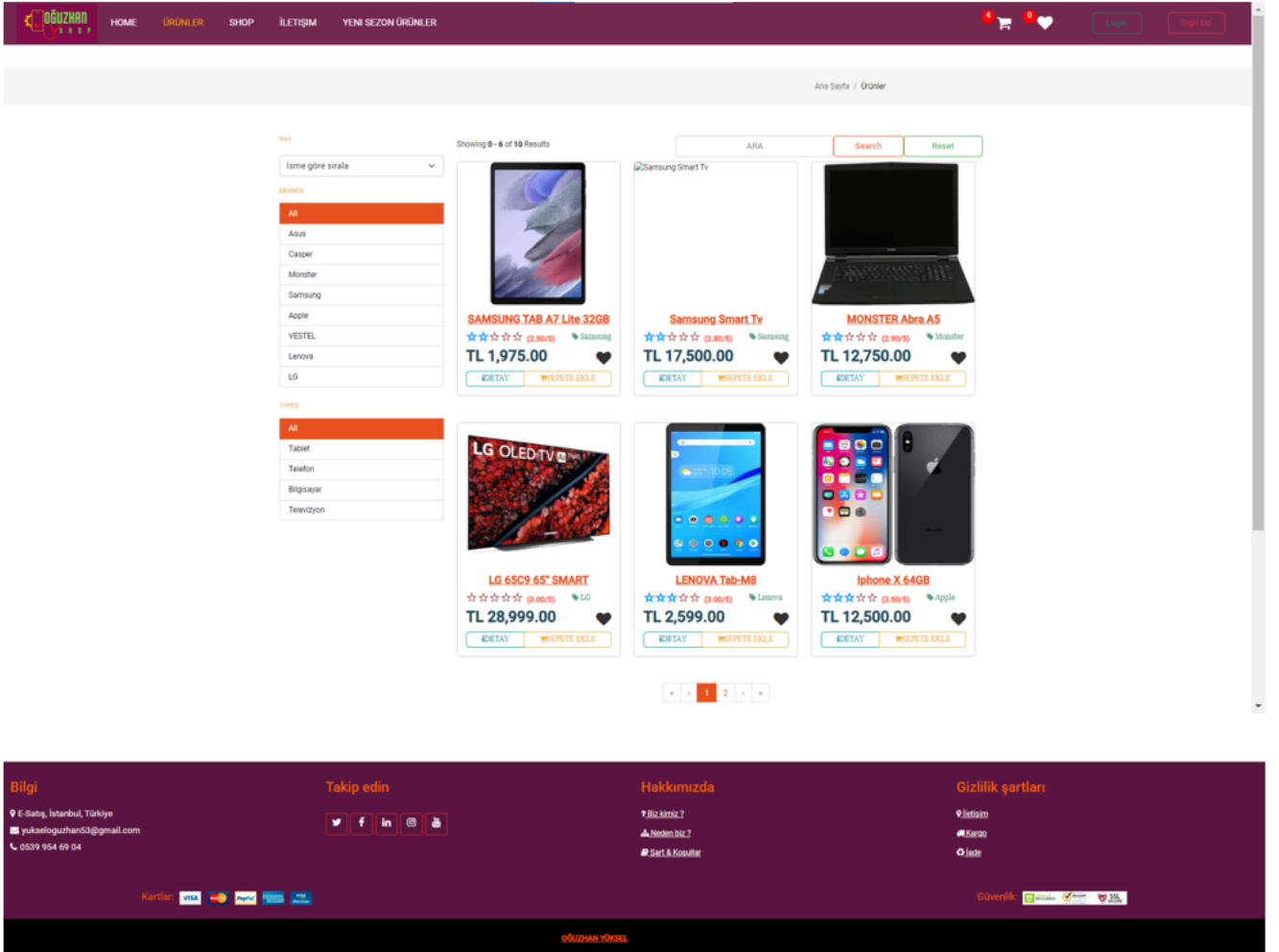
  }

  getCheapProducts() {
    return this.http.get<ICheapProducts>(this.baseUrl + "Products/cheap-products");
  }
}
```

Şekil 5.9. [home.service.ts](#) dosyasının oluşturulması



Şekil 5.11. Proje anasayfa görünümü



Şekil 5.13. Ürünler sayfası görünümü

İsme göre sırala

BRANDS

All

Asus

Casper

Monster

Samsung

Apple

VESTEL

Lenova

LG

TYPES

All

Tablet

Telefon

Bilgisayar

Televizyon

İphone 11 128gb
☆☆☆☆☆ (0.00/5) Apple
TL 15,499.00

CASPER Excalibure G77 I7
☆☆☆☆☆ (0.00/5) Casper
TL 18,654.00

ASUS ME70C 8GB
☆☆☆☆☆ (0.00/5) ASUS
TL 3,210.00

APPLE MacBook PRO M2
☆☆☆☆☆ (2.00/5) Apple
TL 37,999.00

İDETAY SEPETE EKLE

İDETAY SEPETE EKLE

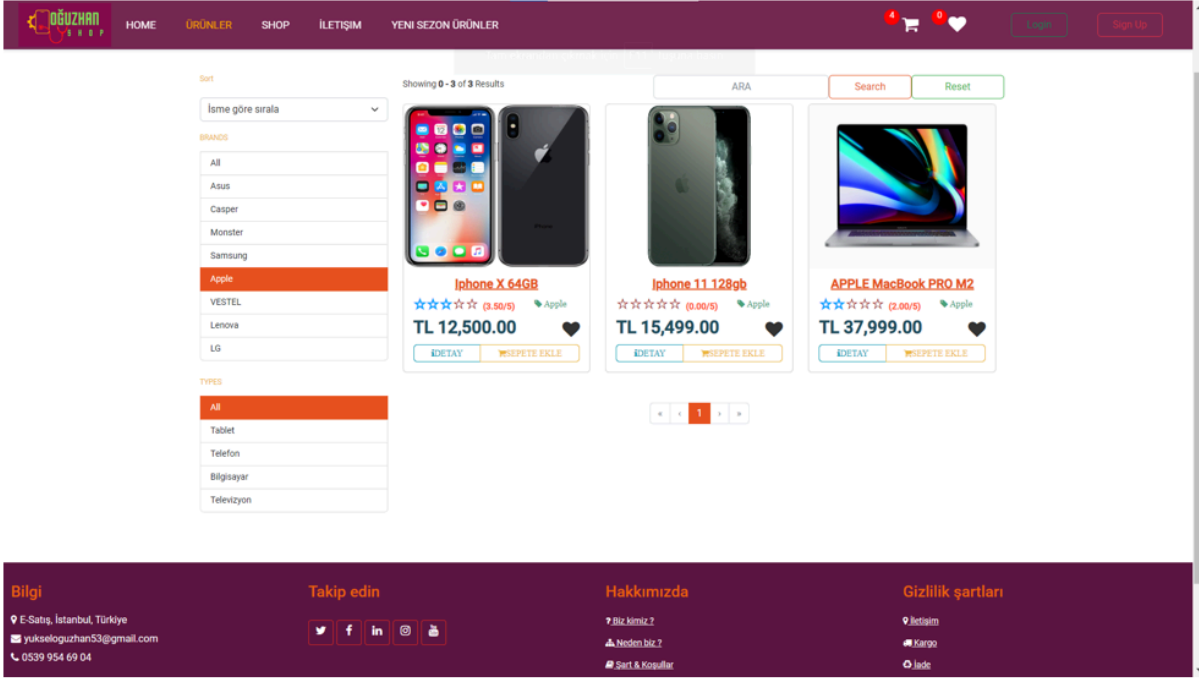
İDETAY SEPETE EKLE

İDETAY SEPETE EKLE

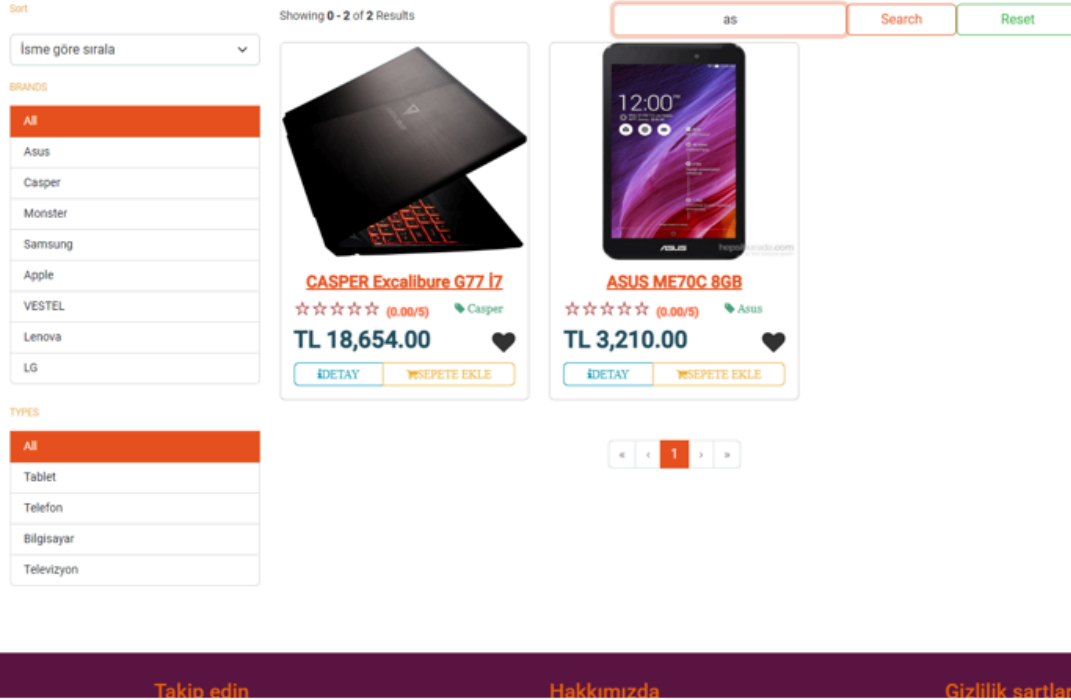
İDETAY SEPETE EKLE

1 2

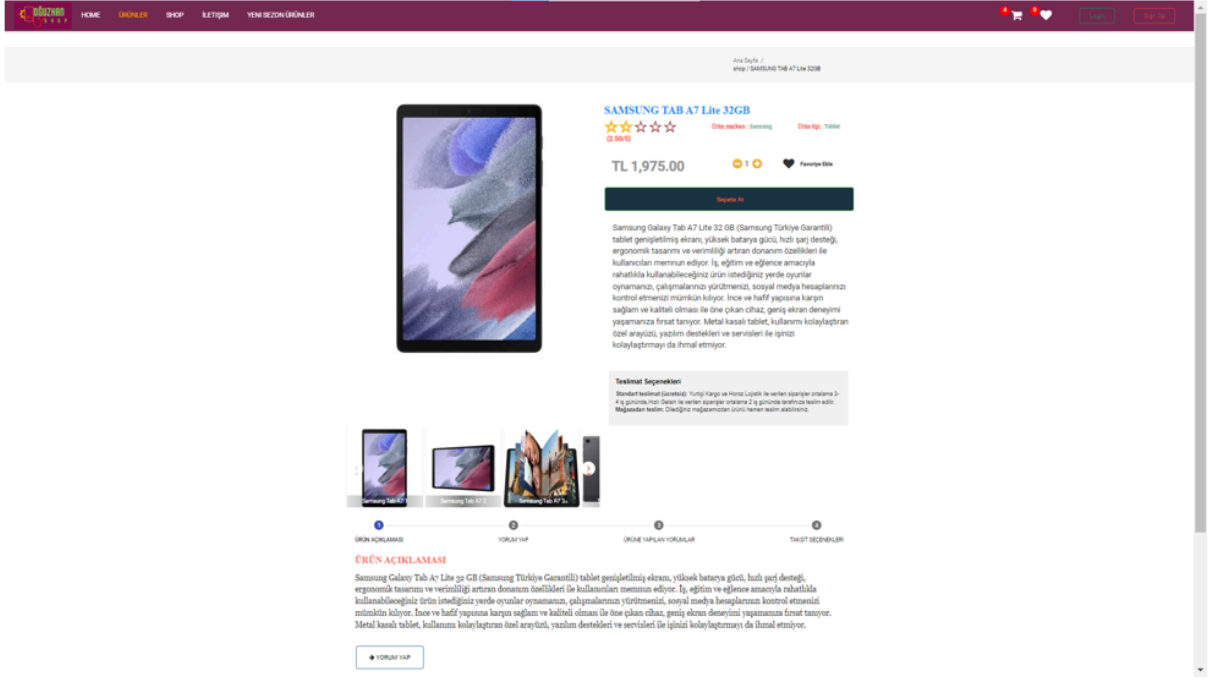
Şekil 5.14. Sayfalama eklenmesi



Şekil 5.15. Projede ürün filtrelemenin gösterilmesi

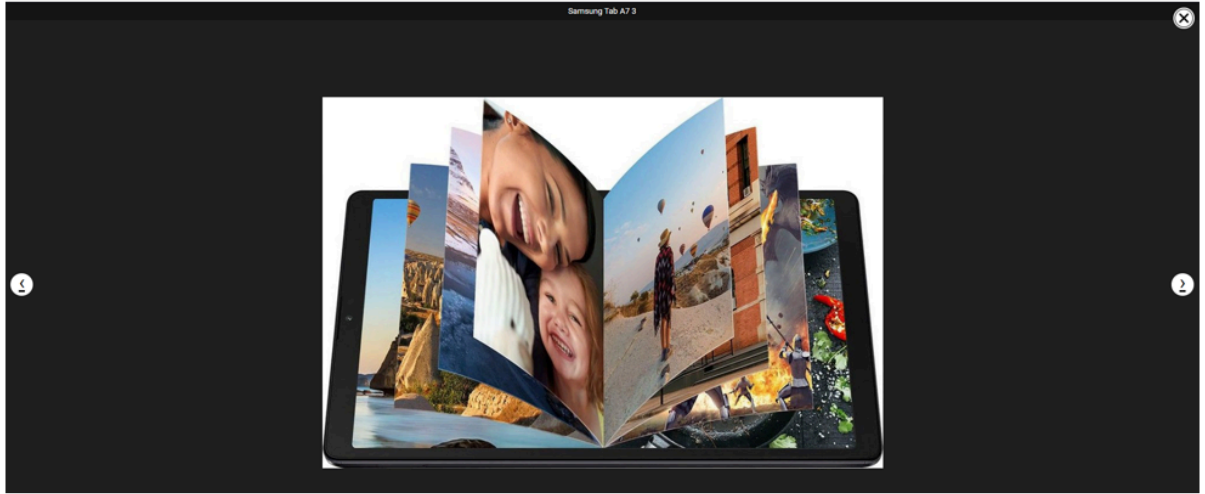


Şekil 5.16. Proje ürün arama işleminin gösterilmesi



Şekil 5.17. Ürün detay sayfası görünümü

Ürüne ait resimleri görüp daha yakından inceleyebilir.



Şekil 5.18. Ürün resimleri görünümü

Ürün detayına girdiğinde ürüne ait yorumları görebilir.

ÜRÜN AÇIKLAMASI 2 YORUM YAP 3 ÜRÜNE YAPILAN YORUMLAR 4 TAKSİT SEÇENEKLERİ

ÜRÜNE YAPILAN YORUMLAR

Test3 - 14 June 2022 - 00 : 00
★★★★★
Test3 description

Test4 - 01 January 1 - 00 : 00
★★★★★
Test4description

Test6 - 01 January 1 - 00 : 00
★★★★★
Test6description

test19 - 04 December 2022 - 00 : 47
★★★★★
test19

→ TAKSİT SEÇENEKLERİ

Şekil 5.19. Ürüne yapılan yorumlar

ÜRÜN AÇIKLAMASI 2 YORUM YAP 3 ÜRÜNE YAPILAN YORUMLAR 4 TAKSİT SEÇENEKLERİ

YORUM YAP

Kullanıcı Adı

Yorum

GÖNDER

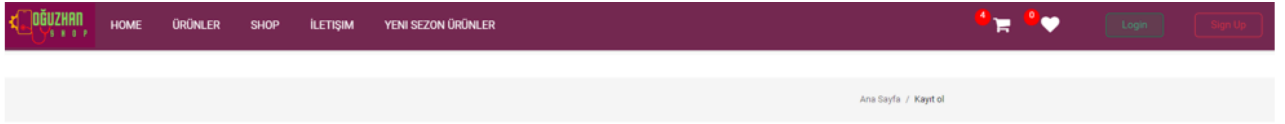
ÜRÜNÜ DEĞERLENDİR

★★★★★

GÖNDER

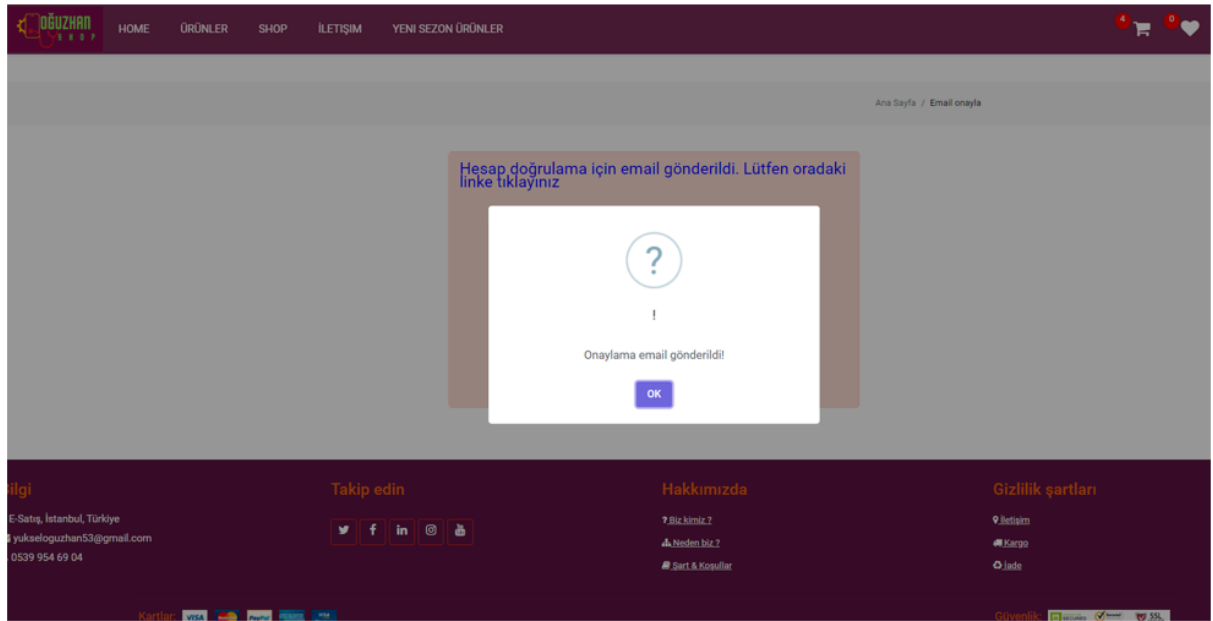
SONUC : 0

Şekil 5.20. Ürünü değerlendirme ve yorum yapma



The screenshot shows a mobile registration form titled 'KAYIT OL'. The form is set against a dark purple background. It includes a user icon at the top, followed by input fields for 'Ad:' (Hasan), 'Kullanıcı adı:' (hasanhasan), 'Email Adresi:' (hasan@gmail.com), and 'Şifre:'. Below the fields is an orange 'GİRİŞ YAP' button and a link for 'OĞUZHAN SHOP'.

Şekil 5.22. Register sayfa görünümü



Şekil 5.23. Hesap onay email'ı gönderilmesi

🕒 10 dakikalık e-posta'ya hoşgeldiniz.

🕒 E-posta adresiniz 07:51 içinde sona erecek.

xpj82115@xcoxc.com

📄 Panoya kopyala.

- 🔄 Sayfayı yenile.
- 🕒 10 dakika daha ver!
- 📧 Başka bir e-posta adresi ver. (Bu adres sona erecek)

📧 Mesajı oku!

Raporla! Düz metin URL

Confirm your email address



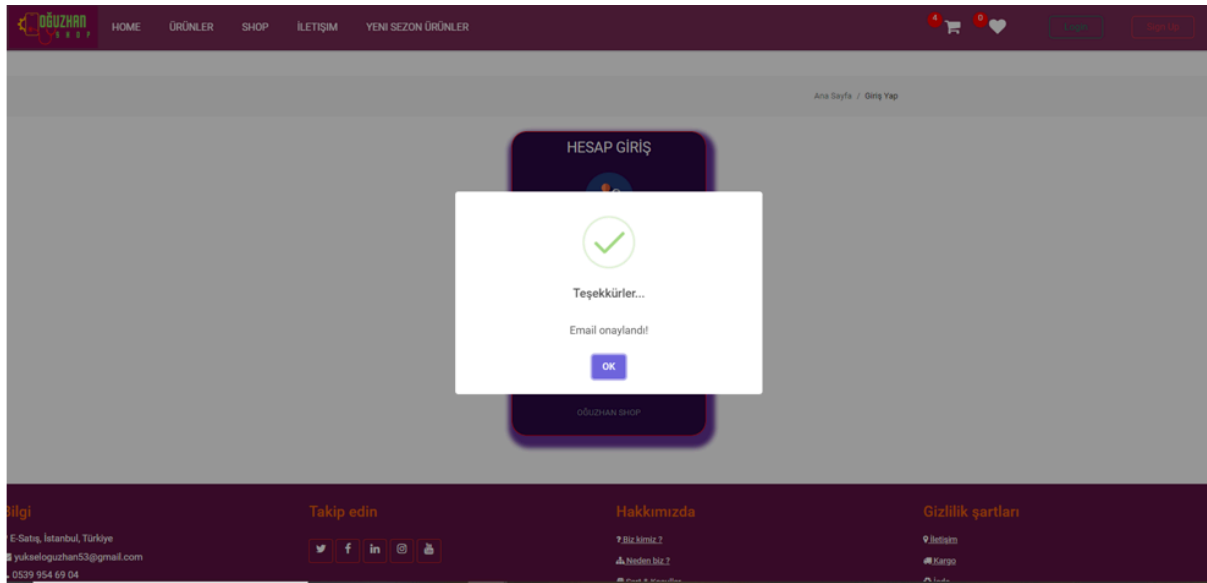
yukseloguzhan53@gmail.com (yukseloguzhan53@gmail.com) 1 dakika önce

Alıcı: xpj82115@xcoxc.com

Lütfen hesabınızı onaylamak için aşağıdaki linke tıklayınız:

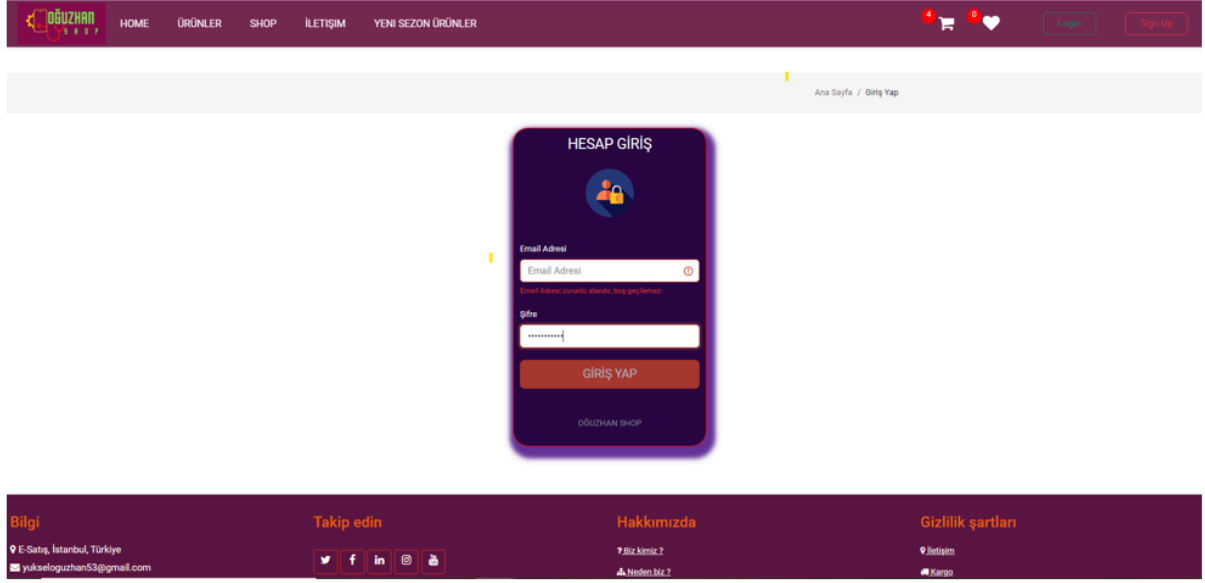
<http://localhost:4200/account/confirm-email?userid=aad00654-bfbd-4f67-bce8-fba7d15c2f07&token=CfdJ8A9bKRJ9VUJLsJVoeiugQVFqkTjBwWnQExS9L16PjqltbEsluxoU5d5fbXXo8BqznBz%2blgNO45YF8nFH0x3qdZzV3bo3U3%2bEFYnhgEYJ5T84i4PUFB09rIrFB%2fdDHE57IGc068%2byta4EKKk4Vdf%2fzl.jwFz4bbS%2fF%2f0X6cj%2fE20Mcx4gbgrO61ykRq9NXT5emVTNQCzEQGz8OHh2r9d67euygYNdmmvA6ZEF%2fQf7HolnepUwq00ByjaLcQD3gwl%2fQp%3d%3d>

Şekil 5.24. Kullanıcının hesabına onay linki gelmesi

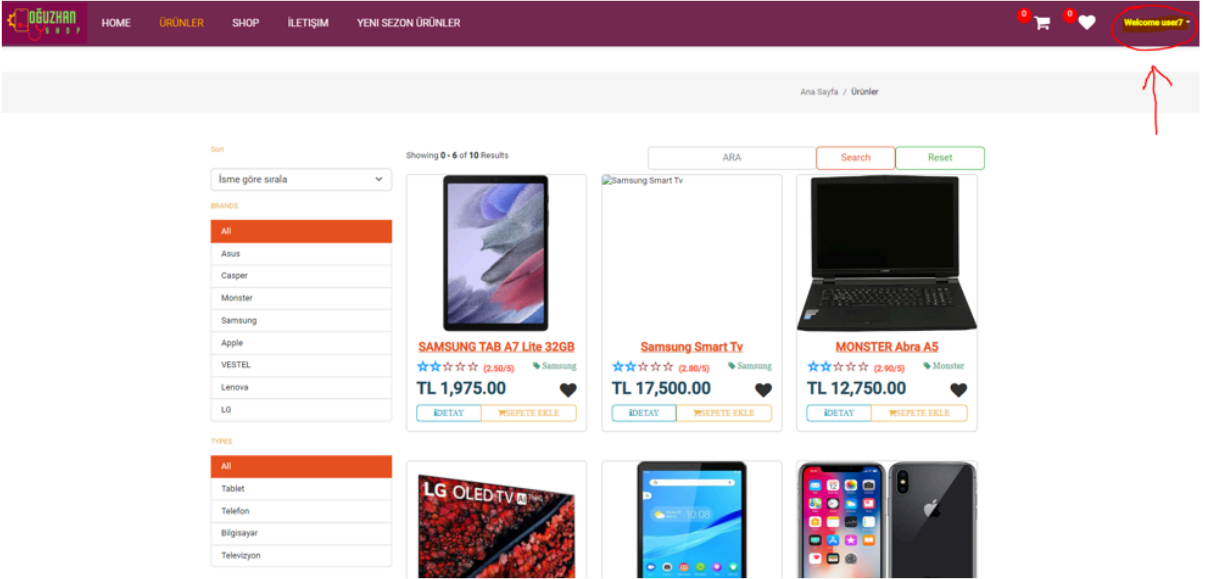


Şekil 5.25. Kullanıcının hesabını onaylaması

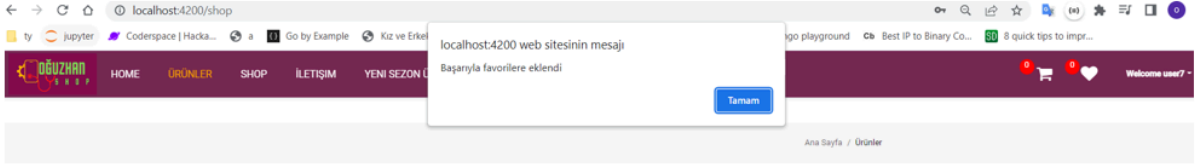
Daha sonra üye bilgilerini (email ve şifre) girdikten sonra sisteme girebilir.



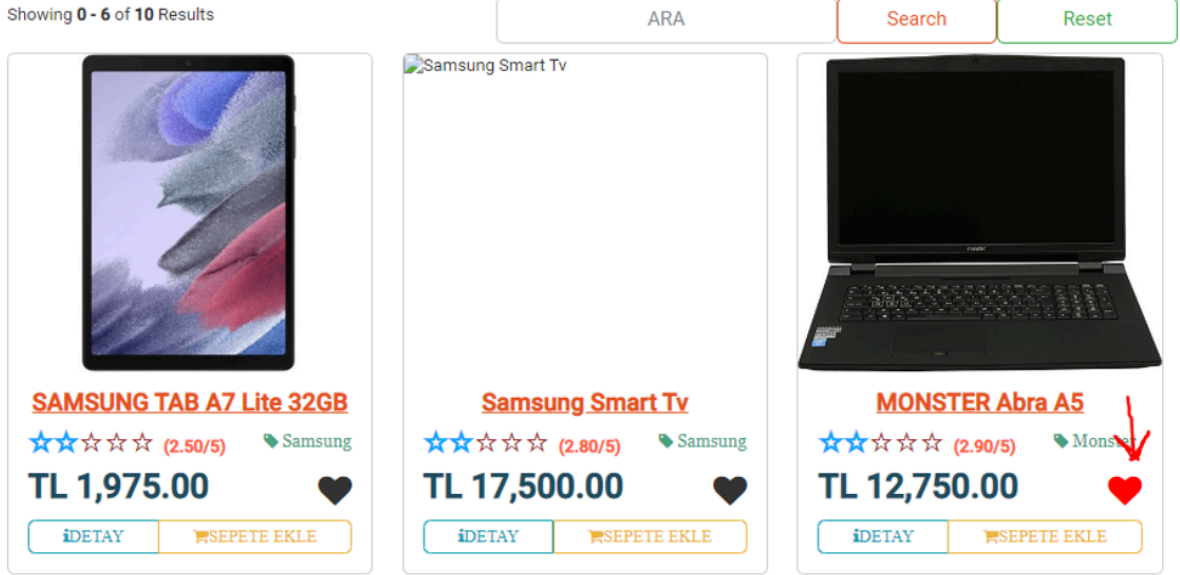
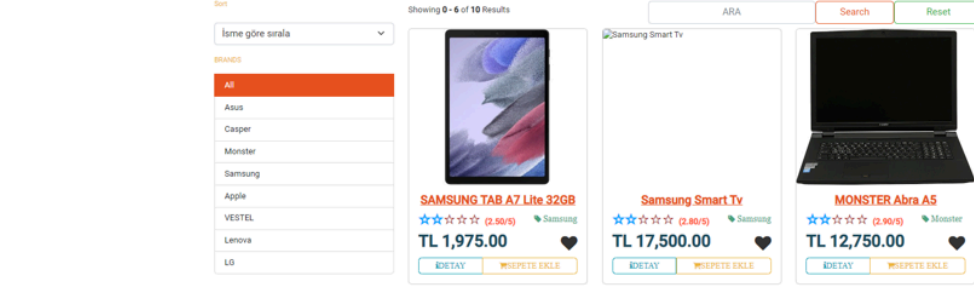
Şekil 5.26. Hesap girişi ekranı



Şekil 5.27. Kullanıcının giriş yapması



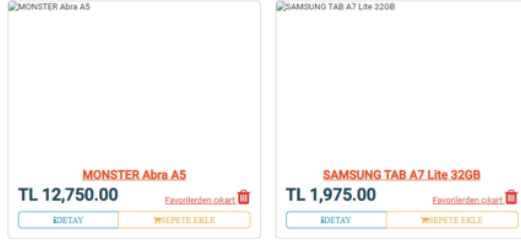
Şekil 5.29. Ürünü favorilere ekleme



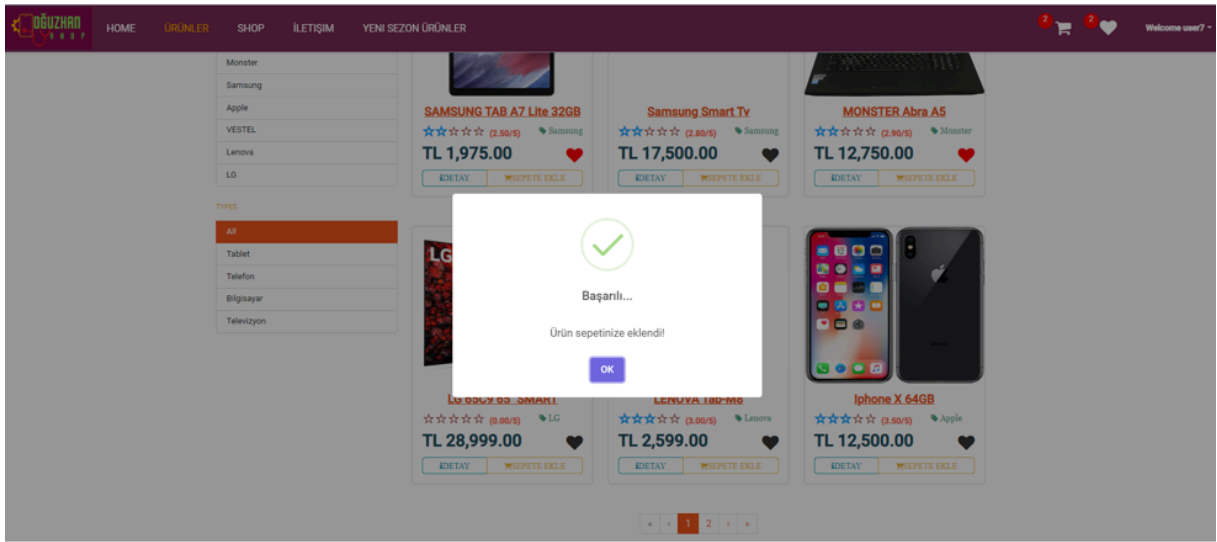
Şekil 5.30. Favorilere eklenmiş ürün görünümü







Şekil 5.31. Kullanıcının favori ürün sayısı



Şekil 5.32. Kullanıcının favori listesi



Şekil 5.33. Kullanıcının ürünü sepetine eklemesi

PRODUCT	PRICE	QUANTITY	TOTAL	REMOVE
	\$1,975.00	1	\$1,975.00	
	\$12,500.00	1	\$12,500.00	

SİPARİŞ DETAY

Ödeme sırasında yapılan seçimlere bağlı olarak nakliye ücretleri eklenecektir

Sipariş Dip Toplam	\$14,475.00
Nakliye Ücreti	\$0.00
Toplam	\$14,475.00

[Ödeme Yap](#)

Şekil 5.34. Kullanıcının sepeti görünümü

ÖZÜZHAN HOME ÜRÜNLER SHOP İLETİŞİM YENİ SEZON ÜRÜNLER Welcome user7

Ana Sayfa / Ödeme

ADDRESS DELIVERY REVIEW PAYMENT SİPARİŞ DETAY

Address Informations

[Save as default address](#)

Oğuzhan Yüksel

Bahçelievler, 289/1. Sk. No:9, 06630 Gölbaşı/Ankara Gölbaşı

Türkiye 34764

[Back to Basket](#) [Go to Delivery](#)

Sipariş Dip Toplam \$14,475.00

Nakliye Ücreti \$0.00

Toplam \$14,475.00

Bilgi E-Satış, İstanbul, Türkiye
yukseoguzhan53@gmail.com
0539 954 69 04

Takip edin

Hakkımızda Biz kimiz? Neden biz? İletişim & Koşullar

Gizlilik şartları İletişim Kargo İade

Kartlar: VISA, Mastercard, Apple Pay, Google Pay, Bankalar

Güvenlik: SSL, PCI DSS, 2FA, 3D Secure

ÖZÜZHAN YÜKSEL

Şekil 5.36. Sipariş vermek için adres bilgileri girilmesi

Kullanıcı aşağıdaki gibi kargo seçeneklerinden uygun olanı seçer.

Ana Sayfa / Ödeme

ADDRESS DELIVERY REVIEW PAYMENT SIPARIŞ DETAY

Hızlı kargo - \$55.00
Test1
 Normal teslimat - \$10.00
Test3
[Back to Address](#)

30 dk da kapıda - \$120.00
Test2
 Teslimat noktası - \$25.00
Test4
[Go to Review](#)

Ödeme sırasında yapılan seçimlere bağlı olarak nakliye ücretleri eklenecektir

Sipariş Dış Toplam	\$14,475.00
Nakliye Ücreti	\$0.00
Toplam	\$14,475.00

Şekil 5.38. Kullanıcının kargo seçeneği seçme ekranı

Sipariş vermek isteyen kullanıcı kart bilgilerini girer ve siparişini vermiş olur.

Ana Sayfa / Ödeme

ADDRESS DELIVERY REVIEW PAYMENT SIPARIŞ DETAY

John Doe

5528790000000008

12 2034 123

[Back to Review](#) [Submit Order](#)

Ödeme sırasında yapılan seçimlere bağlı olarak nakliye ücretleri eklenecektir

Sipariş Dış Toplam	\$3,950.00
Nakliye Ücreti	\$0.00
Toplam	\$3,950.00

Şekil 5.39. Kullanıcının kart bilgilerini girme ekranı

Ödemeleri control için IYZICO ödeme servisini kullandım.

iyzico
a PayU company

Giriş Yap

kuzeyfirtina2@gmail.com

[Destek](#) [Şifremi Unuttum](#)

Henüz hesabın yok mu?

Şekil 5.40. İyzico'ya giriş yapma

iyzico sandbox TR

Güncel Durum

İşlemler

Tüm İşlemler

İadeler

İptaller

Şüpheli İşlemler

Validasyon Hataları

Harcama İtirazları

Para Gönderimleri

Faturalar

iyziLink

Korumalı Alışveriş

Raporlar

Tüm İşlemler

Göster 20 Toplam 13 DISA AKTAR

Tarih Aralığı: 20.10.2022 - 19.12.2022 Filtrele Detay Elde

Tarih Aralığı: 20.10.2022 - 19.12.2022 Para Birimi: TRY HEPSİNİ TEMİZLE

Ödeme Numarası	Oluşturma Tarihi	Tahsil Edilen Tutar	Conversation ID	Ödeme Fazı	Alıcı Adı	Alıcı
18652794	19.12.2022 16:27:23	₺14.845,00	efda8d4a-b491-4cee-b229-985b47a22e1d	Ödeme (Auth)	Oğuzhan	YÜ
18652684	19.12.2022 16:18:44	₺41.869,00	0995e1d1-2e7e-4591-8f58-afec38f63f1c	Ödeme (Auth)	Oğuzhan	YÜ
18652583	19.12.2022 16:05:35	₺67.369,00	9084db05-ff66-46f3-9dc3-e0d718503a52	Ödeme (Auth)	Oğuzhan	YÜ
18652523	19.12.2022 15:56:39	₺38.055,00	cc28a638-df38-4ddb-9b2e-b6ffd53eb617	Ödeme (Auth)	Oğuzhan	YÜ

Şekil 5.41. Yapılan ödeme listesi

18652794 nolu ödeme sayfası TUM ÖDEMEYİ İADE ET

Ödeme Durumu	Ödeme Tarihi	Tahsil Edilen Tutar	Fraud Denetimi
Başarılı	19.12.2022 16:27:23	₺14.845,00	Düşük Fraud İhtimali İşlem

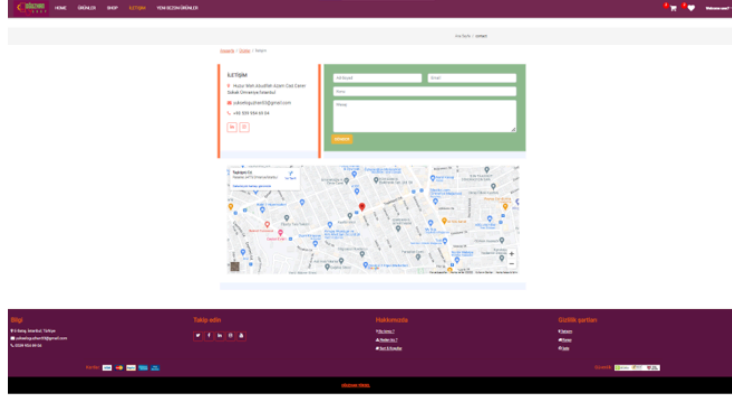
DETAYI GÖSTER

Alıcı Bilgileri

Ad	Soyad	E-posta	Cep Telefonu Numarası
Oğuzhan	Yüksel	eff13462@nezid.com	+905350000000
Alıcı Numarası	IP	Kayıt Tarihi	Son Giriş Tarihi
BY789	85.34.78.112	21.04.2013 15:12:09	05.10.2015 12:43:35
Kayıtlı Adres	Ülke	Şehir	TC Kimlik Numarası
Nidakule Göztepe, Merdivenköy Mah. Bora Sok. No:1	Turkey	Gölbahşı	74300864791

Şekil 5.42. Yapılan ödeme detayını görme

Kullanıcıların istek ve taleplerini iletmesi için iletişim sayfası oluşturdum.



Şekil 5.43. İletişim sayfası tasarımı