

# **Dokumentacija ASP .NET CORE**

## **Projekat iz ASP-a**

# Opis projekta

Projekat sadrzi odradjenu:

**1) Arhitekturu**

**2) Serversku Validaciju**

**3) Pretragu**

**4) Optimalan dizajn baze podata + code first pristup**

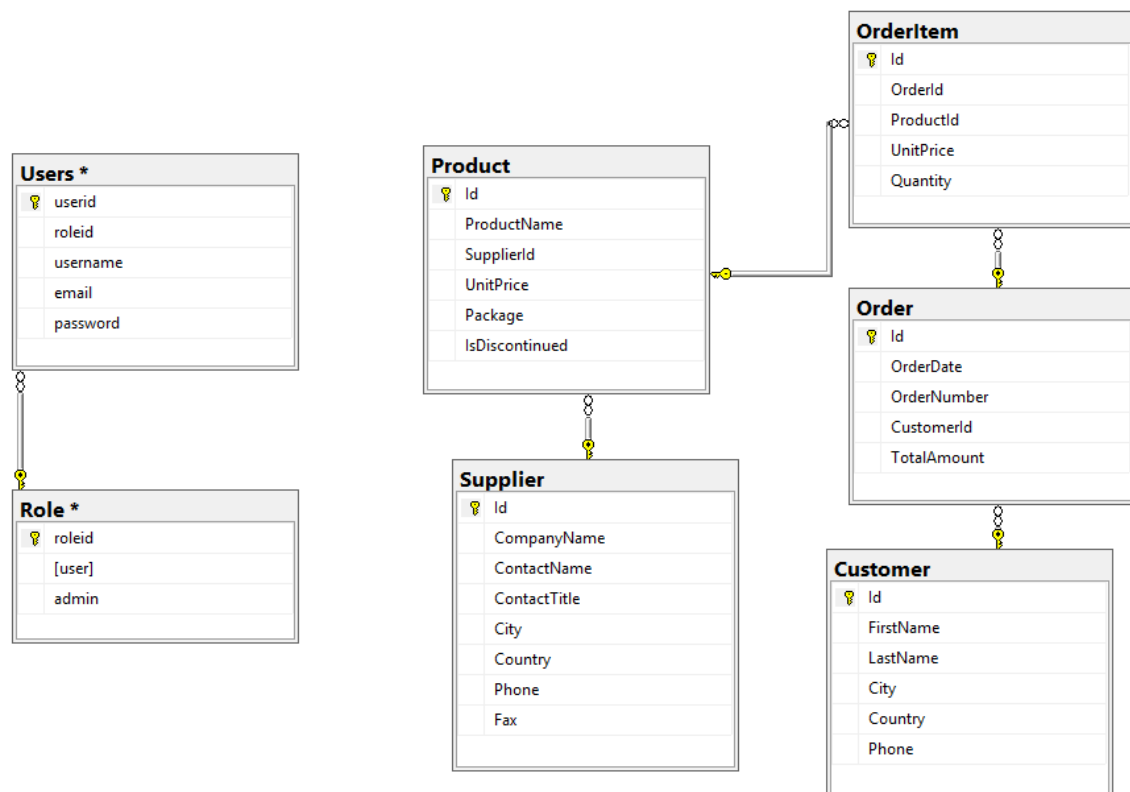
**5) Ispravno vraćanje statusnih kodova uz HTTP response**

**6) Web MVC crud stranice za dva ključna entiteta**

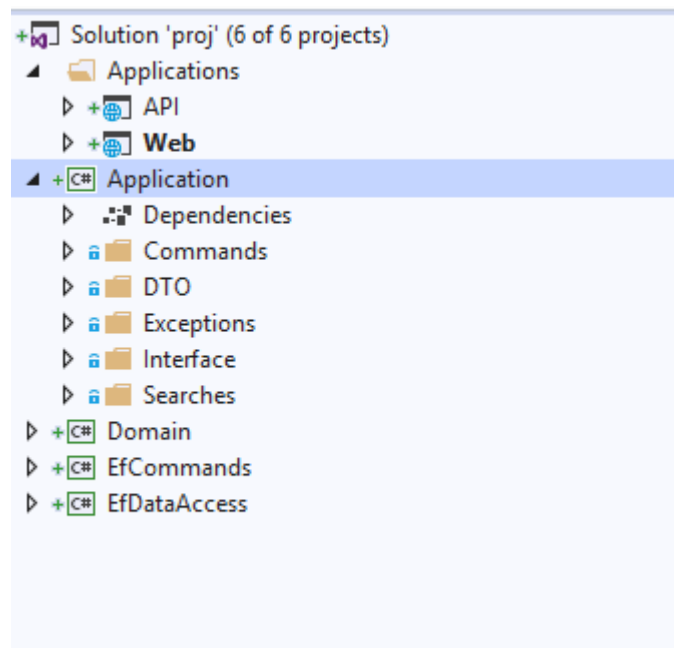
Koristi se Factory database.

Dijagram databaze

---



Solution explorer projekta:



### **Lista kontrolera u API-ju:**

- CustomerController
- ProductController
- OrderController
- UsersController
- SupplierController

### **Lista kontrolera u Web-u:**

- UsersController
- ProductController

Za svaki od kljucnih kontrolera postoje view-ovi za:

- 1) Create
- 2) Edit
- 3) Delete
- 3) Details
- 4) Index

# KOD

## EfDataAccess projekat

### ProjContext.cs

```
using System;
using EfDataAccess.Configurations;
using Microsoft.EntityFrameworkCore;

namespace EfDataAccess
{
    public partial class ProjContext : DbContext
    {
        public virtual DbSet<Customer> Customer { get; set; }
        public virtual DbSet<Order> Order { get; set; }
        public virtual DbSet<OrderItem> OrderItem { get; set; }
        public virtual DbSet<Product> Product { get; set; }
        public virtual DbSet<Role> Role { get; set; }
        public virtual DbSet<Supplier> Supplier { get; set; }
        public virtual DbSet<Users> Users { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)
        {
            optionsBuilder.UseSqlServer(@"Data Source=Data
Source=.\SQLEXPRESS;Initial Catalog=Factory;Integrated Security=True");
        }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.ApplyConfiguration(new CustomerConfiguration());
            modelBuilder.ApplyConfiguration(new OrderConfiguration());
            modelBuilder.ApplyConfiguration(new OrderItemConfiguration());
            modelBuilder.ApplyConfiguration(new ProductConfiguration());
            modelBuilder.ApplyConfiguration(new RoleConfiguration());
            modelBuilder.ApplyConfiguration(new SupplierConfiguration());
            modelBuilder.ApplyConfiguration(new UserConfiguration());
        }
    }
}
```

# EfCommands projekt

## Customers folder

### EfAddCustomerCommand.cs

```
using Application.Commands.Customers;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.Customers
{
    public class EfAddCustomerCommand : IAddCustomerCommand
    {
        private readonly ProjContext _context;

        public EfAddCustomerCommand(ProjContext context)
        {
            _context = context;
        }

        public void Execute(CustomerDTO request)
        {
            if (_context.Customer.Any(p => p.Id == request.Id))
            {
                throw new EntityAlreadyExistsException();
            }

            _context.Customer.Add(new Customer
            {
                FirstName = request.FirstName,
                LastName = request.LastName,
                City = request.City,
                Country = request.Phone
            });
        }
    }
}
```

```

        _context.SaveChanges();
    }

}

```

### **EfDeleteUserCommand.cs**

```

using Application.Commands.User;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Text;

namespace EfCommands.Customers
{
    public class EfDeleteUserCommand : IDeleteUserCommand
    {
        private readonly ProjContext context;

        public void Execute(int request)
        {
            var customer = context.Customer.Find(request);

            if (customer == null)
                throw new EntityNotFoundException();

            context.Customer.Remove(customer);

            context.SaveChanges();
        }
    }
}

```

### **EfEditUserCommand.cs**

```

using Application.Commands.Customers;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.Customers
{
    public class EfEditUserCommand : IEditCustomerCommand
    {
        private readonly ProjContext context;

        public EfEditUserCommand(ProjContext context)
        {
            this.context = context;
        }

        public void Execute(CustomerDTO request)
        {
            var customers = context.Customer.Find(request.Id);
            if (customers == null)
            {
                throw new EntityNotFoundException();
            }
            if (customers.Id != request.Id)
            {
                if (context.Customer.Any(p => p.Id == request.Id))
                {
                    throw new EntityAlreadyExistsException();
                }
                customers.LastName = request.LastName;

                context.SaveChanges();
            }
        }
    }
}

```

### **EfGetUserCommand.cs**

```

using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;

```

```

using System;
using System.Collections.Generic;
using System.Text;

namespace EfCommands.User
{
    public class EfGetUserCommand : IGetUserCommand
    {
        private readonly ProjContext context;

        public EfGetUserCommand(ProjContext context)
        {
            this.context = context;
        }

        public UserDTO Execute(int request)
        {
            var users = context.Users.Find(request);

            if (users == null)
                throw new EntityNotFoundException();

            return new UserDTO
            {
                Userid = users.Userid,
                Username = users.Username,
                Email = users.Email
            };
        }
    }
}

```

### **EfGetUsersCommand.cs**

```

using Application.Commands.User;
using Application.DTO;
using Application.Interface;
using Application.Searches;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```



```

namespace EfCommands.User
{
    public class EfGetUsersCommand : IGetUsersCommand
    {
        private readonly ProjContext _context;

        public EfGetUsersCommand(ProjContext context)
        {
            _context = context;
        }

        public IEnumerable<UserDTO> Execute(UserSearch request)
        {
            var query = _context.Users.AsQueryable();

            return query.Select(p => new UserDTO
            {
                Username = request.Username,
                Email = request.Email
            });
        }

        IEnumerable<UserDTO> ICommand<UserSearch,
        IEnumerable<UserDTO>>.Execute(UserSearch request)
        {
            throw new NotImplementedException();
        }
    }
}

```

## Orders folder

### EfAddOrderCommand.cs

```

using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using Application.Interface.Orders;
using EfDataAccess;
using System;

```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.Orders
{
    public class EfAddOrderCommand : IAddOrderCommand
    {
        private readonly ProjContext _context;

        public EfAddOrderCommand(ProjContext context)
        {
            _context = context;
        }

        public void Execute(OrderDTO request)
        {
            if (_context.Order.Any(p => p.OrderNumber == request.OrderNumber))
            {
                throw new EntityAlreadyExistsException();
            }

            _context.Order.Add(new Order
            {
                OrderDate = request.OrderDate,
                OrderNumber = request.OrderNumber,
                CustomerId = request.CustomerId,
                TotalAmount = request.TotalAmount
            });
            _context.SaveChanges();
        }
    }
}

```

### **EfDeleteOrderCommand.cs**

```

using Application.Commands.Orders;
using Application.Commands.User;
using Application.Exceptions;
using EfDataAccess;

```

```

using System;
using System.Collections.Generic;
using System.Text;

namespace EfCommands.Orders
{
    public class EfDeleteOrderCommand : IDeleteOrderCommand
    {
        private readonly ProjContext _context;

        public EfDeleteOrderCommand(ProjContext context)
        {
            _context = context;
        }

        public void Execute(int request)
        {
            var order = _context.Order.Find(request);

            if (order == null)
                throw new EntityNotFoundException();

            _context.Order.Remove(order);

            _context.SaveChanges();
        }
    }
}

```

### **EfEditOrderCommand.cs**

```

using Application.Commands.Orders;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.Orders
{

```

```

public class EfEditOrderCommand : IEditOrderCommand
{
    private readonly ProjContext _context;

    public EfEditOrderCommand(ProjContext context)
    {
        _context = context;
    }

    public void Execute(OrderDTO request)
    {
        var order = _context.Order.Find(request.Id);
        if (order == null)
        {
            throw new EntityNotFoundException();
        }
        if (order.OrderNumber != request.OrderNumber)
        {
            if (_context.Order.Any(p => p.OrderNumber == request.OrderNumber))
            {
                throw new EntityAlreadyExistsException();
            }
            order.OrderNumber = request.OrderNumber;

            _context.SaveChanges();
        }
    }
}

```

### **EfGetOrderCommand.cs**

```

using Application.Commands.Orders;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Text;

namespace EfCommands.Orders
{

```

```

public class EfGetOrderCommand : IGetOrderCommand
{
    private readonly ProjContext context;

    public EfGetOrderCommand(ProjContext context)
    {
        this.context = context;
    }

    public OrderDTO Execute(int request)
    {
        var order = context.Order.Find(request);

        if (order == null)
            throw new EntityNotFoundException();

        return new OrderDTO
        {
            Id = order.Id,
            OrderNumber = order.OrderNumber,
            OrderDate = order.OrderDate
        };
    }
}

```

### **EfGetOrdersCommand.cs**

```

using Application.Commands.Customers;
using Application.Commands.Orders;
using Application.Commands.User;
using Application.DTO;
using Application.Searches;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.Orders
{
    public class EfGetOrdersCommand : IGetOrderCommand
    {

```

```

private readonly ProjContext context;

public EfGetOrdersCommand(ProjContext context)
{
    this.context = context;
}

public IEnumerable<OrderDTO> Execute(OrderSearch request)
{
    var query = context.Order.AsQueryable();

    return query.Select(p => new OrderDTO
    {
        Id = request.Id,
        OrderNumber = request.OrderNumber
    });
}

public OrderDTO Execute(int request)
{
    throw new NotImplementedException();
}
}

```

## **Product folder**

### **EfAddProductCommand.cs**

```

using Application.DTO;
using Application.Exceptions;
using Application.Interface;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands

```

```

{
    public class EfAddProductCommand : IAddProductCommand
    {

        private readonly ProjContext _context;

        public EfAddProductCommand(ProjContext context)
        {
            _context = context;
        }

        public void Execute(ProductDTO request) {

            if (_context.Product.Any(p => p.ProductName == request.ProductName)) {
                throw new EntityAlreadyExistsException();
            }

            _context.Product.Add(new Product {
                ProductName=request.ProductName,
                SupplierId = request.SupplierId,
                UnitPrice = request.UnitPrice,
                Package = request.Package,
                IsDiscontinued = request.IsDiscontinued
            });
            _context.SaveChanges();
        }

    }
}

```

### **EfDeleteProductCommand.cs**

```

using Application.Commands;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Text;

namespace EfCommands
{
    public class EfDeleteProductCommand : IDeleteProductCommand
    {
        private readonly ProjContext context;
    }
}

```

```

public void Execute(int request)
{
    var product = context.Product.Find(request);

    if (product == null)
        throw new EntityNotFoundException();

    context.Product.Remove(product);

    context.SaveChanges();
}
}
}

```

### **EfEditProductCommand.cs**

```

using Application.Commands;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands
{
    public class EfEditProductCommand : IEditProductCommand
    {
        private readonly ProjContext context;

        public EfEditProductCommand(ProjContext context)
        {
            this.context = context;
        }

        public void Execute(ProductDTO request)
        {
            var product = context.Product.Find(request.Id);
            if (product == null) {
                throw new EntityNotFoundException();
            }
            if (product.ProductName != request.ProductName) {

```



```

        if (context.Product.Any(p => p.ProductName == request.ProductName)) {
            throw new EntityAlreadyExistsException();
        }
        product.ProductName = request.ProductName;

        context.SaveChanges();
    }
}
}
}

```

### **EfGetProductCommand.cs**

```

using Application.Commands;
using Application.DTO;
using Application.Searches;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands
{
    public class EfGetProductCommand : IGetProductsCommand
    {
        private readonly ProjContext _context;

        public EfGetProductCommand(ProjContext context)
        {
            _context = context;
        }

        public IEnumerable<ProductDTO> Execute(ProductSearch request)
        {
            var query = _context.Product.AsQueryable();

            return query.Select(p => new ProductDTO {
                ProductName=p.ProductName,
                SupplierId = p.SupplierId,
                UnitPrice = Convert.ToInt32(p.UnitPrice),
                Package = p.Package,
                IsDiscontinued = p.IsDiscontinued
            });
        }
    }
}

```

```
    }  
  
    }  
}
```

### **EfGetProducttCommand.cs**

```
using Application.Commands;  
using Application.DTO;  
using Application.Exceptions;  
using EfDataAccess;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace EfCommands  
{  
    public class EfGetProducttCommand : IGetProductCommand  
    {  
        private readonly ProjContext context;  
  
        public EfGetProducttCommand(ProjContext context)  
        {  
            this.context = context;  
        }  
  
        public ProductDTO Execute(int request)  
        {  
            var produkt = context.Product.Find(request);  
  
            if (produkt == null)  
                throw new EntityNotFoundException();  
  
            return new ProductDTO  
            {  
                Id = produkt.Id,  
                ProductName = produkt.ProductName  
            };  
        }  
    }  
}
```

```
}
```

## SupplierEF folder

### EfAddSupplierCommand.cs

```
using Application.Commands.Supplier;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.SupplierEF
{
    public class EfAddSupplierCommand : IAddSupplierCommand
    {
        private readonly ProjContext _context;

        public EfAddSupplierCommand(ProjContext context)
        {
            _context = context;
        }

        public void Execute(SupplierDTO request)
        {
            if (_context.Supplier.Any(p => p.Id == request.Id))
            {
                throw new EntityAlreadyExistsException();
            }

            _context.Supplier.Add(new Supplier
            {
                Id = request.Id,
                CompanyName = request.CompanyName,
                City = request.City
            });
            _context.SaveChanges();
        }
    }
}
```

```
}  
}
```

### **EfDeleteSupplierCommand.cs**

```
using Application.Commands.Supplier;  
using Application.Commands.User;  
using Application.Exceptions;  
using EfDataAccess;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace EfCommands.SupplierEF  
{  
    public class EfDeleteSupplierCommand : IDeleteSupplierCommand  
    {  
        private readonly ProjContext _context;  
  
        public EfDeleteSupplierCommand(ProjContext context)  
        {  
            _context = context;  
        }  
  
        public void Execute(int request)  
        {  
            var supplier = _context.Supplier.Find(request);  
  
            if (supplier == null)  
                throw new EntityNotFoundException();  
  
            _context.Supplier.Remove(supplier);  
  
            _context.SaveChanges();  
        }  
    }  
}
```

## EfEditSupplierCommand.cs

```
using Application.Commands.Supplier;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.SupplierEF
{
    public class EfEditSupplierCommand : IEditSupplierCommand
    {
        private readonly ProjContext _context;

        public EfEditSupplierCommand(ProjContext context)
        {
            _context = context;
        }

        public void Execute(SupplierDTO request)
        {
            var supplier = _context.Supplier.Find(request.Id);
            if (supplier == null)
            {
                throw new EntityNotFoundException();
            }
            if (supplier.Id != request.Id)
            {
                if (_context.Supplier.Any(p => p.Id == request.Id))
                {
                    throw new EntityAlreadyExistsException();
                }
                supplier.Id = request.Id;
            }
            _context.SaveChanges();
        }
    }
}
```

## **EfGetSupplierCommand.cs**

```
using Application.Commands.Supplier;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Text;

namespace EfCommands.SupplierEF
{
    public class EfGetSupplierCommand : IGetSupplierCommand
    {
        private readonly ProjContext context;

        public EfGetSupplierCommand(ProjContext context)
        {
            this.context = context;
        }

        public SupplierDTO Execute(int request)
        {
            var supplier = context.Supplier.Find(request);

            if (supplier == null)
                throw new EntityNotFoundException();

            return new SupplierDTO
            {
                Id = supplier.Id,
                CompanyName = supplier.CompanyName,
                ContactTitle = supplier.ContactTitle
            };
        }
    }
}
```

## **EfGetSuppliersCommand.cs**

```
using Application.Commands.Customers;
using Application.Commands.Supplier;
using Application.Commands.User;
using Application.DTO;
using Application.Searches;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.SupplierEF
{
    public class EfGetSuppliersCommand : IGetSuppliersCommand
    {
        private ProjContext context;

        public EfGetSuppliersCommand(ProjContext context)
        {
            this.context = context;
        }

        public IEnumerable<SupplierDTO> Execute(SupplierSearch request)
        {
            var query = context.Supplier.AsQueryable();

            return query.Select(p => new SupplierDTO
            {
                CompanyName = request.CompanyName,
                ContactName = request.ContactName
            });
        }
    }
}
```

## UserEF folder

### EfAddUserCommand.cs

```
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.UserEF
{
    public class EfAddUserCommand : IAddUserCommand
    {
        private readonly ProjContext _context;

        public void Execute(UserDTO request)
        {
            if (_context.Users.Any(p => p.Username == request.Username))
            {
                throw new EntityAlreadyExistsException();
            }

            _context.Users.Add(new Users
            {
                Username = request.Username,
                Email = request.Email,
                Password = request.Password
            });
            _context.SaveChanges();
        }
    }
}
```



## **EfDeleteUserCommand.cs**

```
using Application.Commands.User;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Text;

namespace EfCommands.UserEF
{
    public class EfDeleteUserCommand : IDeleteUserCommand
    {
        private readonly ProjContext _context;

        public EfDeleteUserCommand(ProjContext context)
        {
            _context = context;
        }

        public void Execute(int request)
        {
            var users = _context.Users.Find(request);

            if (users == null)
                throw new EntityNotFoundException();

            _context.Users.Remove(users);

            _context.SaveChanges();
        }
    }
}
```

## **EfEditUserCommand.cs**

```
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```

namespace EfCommands.UserEF
{
    public class EfEditUserCommand : IEditUserCommand
    {
        private readonly ProjContext _context;

        public EfEditUserCommand(ProjContext context)
        {
            _context = context;
        }

        public void Execute(UserDTO request)
        {
            var users = _context.Users.Find(request.Userid);
            if (users == null)
            {
                throw new EntityNotFoundException();
            }
            if (users.Username != request.Username)
            {
                if (_context.Users.Any(p => p.Username == request.Username))
                {
                    throw new EntityAlreadyExistsException();
                }
                users.Username = request.Username;

                _context.SaveChanges();
            }
        }
    }
}

```

### **EfGetUserCommand.cs**

```

using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Text;

```

```

namespace EfCommands.UserEF
{
    public class EfGetUserCommand : IGetUserCommand
    {
        private readonly ProjContext context;

        public EfGetUserCommand(ProjContext context)
        {
            this.context = context;
        }

        public CustomerDTO Execute(int request)
        {
            var customer = context.Customer.Find(request);

            if (customer == null)
                throw new EntityNotFoundException();

            return new CustomerDTO
            {
                Id = customer.Id,
                LastName = customer.LastName,
                FirstName = customer.FirstName
            };
        }
    }
}

```

### **EfGetUsersCommand.cs**

```

using Application.Commands.Customers;
using Application.Commands.User;
using Application.DTO;
using Application.Searches;
using EfDataAccess;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EfCommands.UserEF
{
    public class EfGetUsersCommand : IGetCustomersCommand

```

```

{

    private ProjContext context;

    public EfGetUsersCommand(ProjContext context)
    {
        this.context = context;
    }

    public IEnumerable<CustomerDTO> Execute(CustomerSearch request)
    {
        var query = context.Customer.AsQueryable();

        return query.Select(p => new CustomerDTO
        {
            FirstName = request.FirstName,
            LastName = request.Lastname

        });

    }
}

```

## Application projekat

### Commands folder

#### Customers subfolder

#### IaddCustomerCommand.cs

```

using Application.DTO;
using Application.Interface;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Customers
{
    public interface IAddCustomerCommand : ICommand<CustomerDTO>

```

```
{  
}  
}
```

### **IdeleteCustomerCommand.cs**

```
using Application.Interface;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands.Customers  
{  
    public interface IDeleteCustomerCommand : ICommand<int>  
    {  
  
    }  
}
```

### **IeditCustomerCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands.Customers  
{  
    public interface IEditCustomerCommand : ICommand<CustomerDTO>  
    {  
    }  
}
```

### **IgetCustomerCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using Application.Searches;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands.Customers  
{  
    public class IGetCustomerCommand : ICommand<int, CustomerDTO>
```

```

    {
        public CustomerDTO Execute(int request)
        {
            throw new NotImplementedException();
        }
    }
}

```

### **IgetCustomersCommand.cs**

```

using Application.DTO;
using Application.Interface;
using Application.Searches;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Customers
{
    public interface IGetCustomersCommand : ICommand<CustomerSearch,
    IEnumerable<CustomerDTO>>
    {

    }
}

```

### **Orders subfolder**

#### **IaddOrderCommand.cs**

```

using Application.DTO;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Interface.Orders
{
    public interface IAddOrderCommand : ICommand<OrderDTO>
    {
    }
}

```

### **IdeleteOrderCommand.cs**

```
using Application.Interface;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Orders
{
    public interface IDeleteOrderCommand : ICommand<int>
    {

    }
}
```

### **IeditOrderCommand.cs**

```
using Application.DTO;
using Application.Interface;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Orders
{
    public interface IEditOrderCommand : ICommand<OrderDTO>
    {
    }
}
```

### **IgetOrderCommand.cs**

```
using Application.DTO;
using Application.Interface;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Orders
{
    public interface IGetOrderCommand : ICommand<int, OrderDTO>
    {
    }
}
```

## **IgetOrdersCommand.cs**

```
using Application.DTO;
using Application.Interface;
using Application.Searches;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Orders
{
    public interface IGetOrdersCommand : ICommand<OrderSearch,
    IEnumerable<OrderDTO>>
    {

    }
}
```

## **Product subfolder**

### **IaddProductCommand.cs**

```
using Application.DTO;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Interface
{
    public interface IAddProductCommand : ICommand<ProductDTO>
    {
    }
}
```

### **IdeleteProductCommand.cs**

```
using Application.Interface;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands
{
    public interface IDeleteProductCommand : ICommand<int>
    {
    }
}
```



```
}  
}
```

### **IeditProductCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands  
{  
    public interface IEditProductCommand : ICommand<ProductDTO>  
    {  
    }  
}
```

### **IgetProductCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands  
{  
    public interface IGetProductCommand : ICommand<int, ProductDTO>  
    {  
    }  
}
```

### **IgetProductsCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using Application.Searches;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands  
{
```

```
public interface IGetProductsCommand : ICommand<ProductSearch,
IEnumerable<ProductDTO>>
{

}
}
```

### **Supplier subfolder** **IaddSupplierCommand.cs**

```
using Application.DTO;
using Application.Interface;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Supplier
{
    public interface IAddSupplierCommand : ICommand<SupplierDTO>
    {
    }
}
```

### **IdeleteSupplierCommand.cs**

```
using Application.Interface;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Supplier
{
    public interface IDeleteSupplierCommand : ICommand<int>
    {

    }
}
```

### **IeditSupplierCommand.cs**

```
using Application.DTO;
using Application.Interface;
using System;
using System.Collections.Generic;
```

```

using System.Text;

namespace Application.Commands.Supplier
{
    public interface IEditSupplierCommand : ICommand<SupplierDTO>
    {
    }
}

```

### **IgetSupplierCommand.cs**

```

using Application.DTO;
using Application.Interface;
using Application.Searches;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Supplier
{
    public class IGetSupplierCommand : ICommand<int, SupplierDTO>
    {
        public SupplierDTO Execute(int request)
        {
            throw new NotImplementedException();
        }
    }
}

```

### **IgetSuppliersCommand.cs**

```

using Application.DTO;
using Application.Interface;
using Application.Searches;
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Commands.Supplier
{
    public interface IGetSuppliersCommand : ICommand<SupplierSearch,
    IEnumerable<SupplierDTO>>
    {
    }
}

```

```
}  
}
```

## **User subfolder**

### **IaddUserCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands.User  
{  
    public interface IAddUserCommand : ICommand<UserDTO>  
    {  
    }  
}
```

### **IdeleteUserCommand.cs**

```
using Application.Interface;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands.User  
{  
    public interface IDeleteUserCommand : ICommand<int>  
    {  
  
    }  
}
```

### **IeditUserCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Application.Commands.User
```

```
{  
    public interface IEditUserCommand : ICommand<UserDTO>  
    {  
    }  
}
```

### **IgetUserCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using Application.Searches;  
using System;  
using System.Collections.Generic;  
using System.Text;
```

```
namespace Application.Commands.User  
{  
    public class IGetUserCommand : ICommand<int, UserDTO>  
    {  
        public UserDTO Execute(int request)  
        {  
            throw new NotImplementedException();  
        }  
    }  
}
```

### **IgetUsersCommand.cs**

```
using Application.DTO;  
using Application.Interface;  
using Application.Searches;  
using System;  
using System.Collections.Generic;  
using System.Text;
```

```
namespace Application.Commands.User  
{  
    public interface IGetUsersCommand : ICommand<UserSearch,  
IEnumerable<UserDTO>>  
    {  
    }  
}
```

## **DTO Folder**

### **CustomerDTO.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.DTO
{
    public class CustomerDTO
    {
        public int Id { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string City { get; set; }
        public string Country { get; set; }
        public string Phone { get; set; }
    }
}
```

### **OrderDTO.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.DTO
{
    public class OrderDTO
    {
        public int Id { get; set; }
        public DateTime OrderDate { get; set; }
        public string OrderNumber { get; set; }
        public int CustomerId { get; set; }
        public decimal? TotalAmount { get; set; }
    }
}
```

### **ProductDTO.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
```

```

using System.Text;

namespace Application.DTO
{
    public class ProductDTO
    {
        public int Id { get; set; }
        [Required]
        [MinLength(3,ErrorMessage ="Name must be 3 characters")]
        public string ProductName { get; set; }
        [Required]
        public int SupplierId { get; set; }
        [Required]
        public decimal UnitPrice { get; set; }
        [Required]
        public string Package { get; set; }
        [Required]
        public bool IsDiscontinued
        {
            get; set;
        }
    }
}

```

## SupplierDTO

```

using System;
using System.Collections.Generic;
using System.Text;

namespace Application.DTO
{
    public class SupplierDTO
    {
        public int Id { get; set; }
        public string CompanyName { get; set; }
        public string ContactName { get; set; }
        public string ContactTitle { get; set; }
        public string City { get; set; }
        public string Country { get; set; }
        public string Phone { get; set; }
        public string Fax { get; set; }
    }
}

```

## **UserDTO.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.DTO
{
    public class UserDTO
    {
        public int Userid { get; set; }
        public string Username { get; set; }
        public string Email { get; set; }
        public string Password { get; set; }
    }
}
```

## **Exceptions folder**

### **EntityAlreadyExistsException.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Exceptions
{
    public class EntityAlreadyExistsException : Exception
    {
    }
}
```

### **EntityNotFoundException.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Exceptions
{
    public class EntityNotFoundException : Exception
    {
    }
}
```



## **Interface folder**

### **Icommand.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Interface
{
    public interface ICommand<TRequest>
    {
        void Execute(TRequest request);
    }

    public interface ICommand<TRequest, TResult>
    {
        TResult Execute(TRequest request);
    }
}
```

## **Searches folder**

### **CustomerSearch.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Searches
{
    public class CustomerSearch
    {
        public int Id { get; set; }
        public string FirstName { get; set; }
        public string Lastname { get; set; }
    }
}
```

## **OrderSearch.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Searches
{
    public class OrderSearch
    {
        public int Id { get; set; }
        public string OrderNumber { get; set; }
    }
}
```

## **ProductSearch.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Searches
{
    public class ProductSearch
    {
        public string keyword { get; set; }
        public bool OnlyActive { get; set; }
    }
}
```

## **SupplierSearch.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Searches
{
    public class SupplierSearch
    {
        public int Id { get; set; }
        public string CompanyName { get; set; }
        public string ContactName { get; set; }
    }
}
```

## **UserSearch.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Application.Searches
{
    public class UserSearch
    {
        public string Username { get; set; }
        public string Email { get; set; }
    }
}
```

## **API projekat**

### **CustomerController.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Application.Commands.Customers;
using Application.DTO;
using Application.Exceptions;
using Application.Searches;
using EfDataAccess;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

namespace API.Controllers
{
    public class CustomerController : Controller
    {
        private readonly ProjContext context;
        private readonly IAddCustomerCommand addCustomer;
        private readonly IGetCustomerCommand getCustomer;
        private readonly IEditCustomerCommand editCustomer;
        private readonly IDeleteCustomerCommand deleteCustomer;
        private readonly IGetCustomersCommand getCustomers;
    }
}
```

```
public CustomerController(ProjContext context, IAddCustomerCommand
addCustomer, IGetCustomerCommand getCustomer, IEditCustomerCommand
editCustomer, IDeleteCustomerCommand deleteCustomer, IGetCustomersCommand
getCustomers)
{
    this.context = context;
    this.addCustomer = addCustomer;
    this.getCustomer = getCustomer;
    this.editCustomer = editCustomer;
    this.deleteCustomer = deleteCustomer;
    this.getCustomers = getCustomers;
}
```

```
// GET api/categories
[HttpGet]
public IActionResult Get([FromQuery]CustomerSearch query)
=> Ok(getCustomers.Execute(query));
```

```
// GET api/categories/5
[HttpGet("{id}")]
public IActionResult Get(int id)
{
    try
    {
        var userDTO = getCustomer.Execute(id);
        return Ok(userDTO);
    }
    catch (EntityNotFoundException)
    {
        return NotFound();
    }
}
```

```
// POST api/values
[HttpPost]
public IActionResult Post([FromBody] CustomerDTO dto)
{
    var customer = new Customer
    {
        LastName = dto.LastName
    };
}
```

```

context.Customer.Add(customer);

try
{
    context.SaveChanges();

    return Created("/api/users/" + customer.Id, new CustomerDTO
    {
        Id = customer.Id,
        LastName = customer.LastName,
        FirstName = customer.FirstName
    });
}
catch
{
    return StatusCode(500, "An error has occurred.");
}
}

// PUT api/values/5
[HttpPut("{id}")]
public IActionResult Put(int id, [FromBody] CustomerDTO dto)
{
    var customer = context.Customer.Find(id);

    if (customer == null)
    {
        return NotFound();
    }

    /// if (customer.IsDeleted)
    /// {
    ///     return NotFound();
    /// }

    if (context.Customer.Any(c => c.Id == dto.Id))
    {
        return Conflict("Takav customer vec postoji.");
    }

    customer.Id = dto.Id;

    try
    {

```

```

        context.SaveChanges();
        return NoContent();
    }
    catch (Exception)
    {

        return StatusCode(500, "An error has occurred.");
    }
}

// DELETE api/categories/5
[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    var user = context.Customer.Find(id);

    if (user == null)
    {
        return NotFound();
    }

    /// if (user.IsDeleted)
    ///     return Conflict("Customer je obrisan.");

    ///user.IsDeleted = true;

    try
    {
        context.SaveChanges();
        return NoContent();
    }
    catch (Exception)
    {

        return StatusCode(500);
    }
}
}
}

```

## OrderController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Application.Commands.Orders;
using Application.DTO;
using Application.Exceptions;
using Application.Interface.Orders;
using Application.Searches;
using EfDataAccess;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

namespace API.Controllers
{
    public class OrderController : Controller
    {
        private readonly ProjContext context;
        private readonly IAddOrderCommand addOrder;
        private readonly IGetOrderCommand getOrder;
        private readonly IEditOrderCommand editOrder;
        private readonly IDeleteOrderCommand deleteOrder;
        private readonly IGetOrdersCommand getOrders;

        // GET: Order
        public IActionResult Get([FromQuery]OrderSearch query)
            => Ok(getOrders.Execute(query));

        // GET: Order/Details/5
        public IActionResult Get(int id)
        {
            try
            {
                var orderDTO = getOrder.Execute(id);
                return Ok(orderDTO);
            }
            catch (EntityNotFoundException)
            {
                return NotFound();
            }
        }
    }
}
```

```

[HttpPost]
public IActionResult Post([FromBody] OrderDTO dto)
{
    var order = new Order
    {
        OrderNumber = dto.OrderNumber
    };

    context.Order.Add(order);

    try
    {
        context.SaveChanges();

        return Created("/api/users/" + order.Id, new OrderDTO
        {
            Id = order.Id,
            OrderDate = order.OrderDate,
            OrderNumber = order.OrderNumber
        });
    }
    catch
    {
        return StatusCode(500, "An error has occurred.");
    }
}

// PUT api/values/5
[HttpPut("{id}")]
public IActionResult Put(int id, [FromBody] OrderDTO dto)
{
    var order = context.Order.Find(id);

    if (order == null)
    {
        return NotFound();
    }

    /// if (customer.IsDeleted)
    /// {
    ///     return NotFound();
    /// }

    if (context.Order.Any(c => c.Id == dto.Id))

```



```

    {
        return Conflict("Order je vec unet.");
    }

    order.Id = dto.Id;

    try
    {
        context.SaveChanges();
        return NoContent();
    }
    catch (Exception)
    {

        return StatusCode(500, "An error has occurred.");
    }
}

// DELETE api/categories/5
[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    var order = context.Order.Find(id);

    if (order == null)
    {
        return NotFound();
    }

    /// if (user.IsDeleted)
    ///     return Conflict("Customer je obrisan.");

    ///user.IsDeleted = true;

    try
    {
        context.SaveChanges();
        return NoContent();
    }
    catch (Exception)
    {

        return StatusCode(500);
    }
}

```

```
    }  
  }  
}
```

## **ProductController.cs**

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Threading.Tasks;  
using Application.Commands;  
using Application.DTO;  
using Application.Exceptions;  
using Application.Interface;  
using Application.Searches;  
using EfDataAccess;  
using Microsoft.AspNetCore.Http;  
using Microsoft.AspNetCore.Mvc;  
  
namespace API.Controllers  
{  
  
    public class ProductController : Controller  
    {  
        private readonly ProjContext context;  
        private readonly IAddProductCommand _addProduct;  
        private readonly IGetProductsCommand _getProducts;  
        private readonly IGetProductCommand _getProduct;  
        private readonly IEditProductCommand _editProduct;  
        private readonly IDeleteProductCommand _deleteProduct;  
  
        public ProductController(IGetProductsCommand getProducts, ProjContext  
context, IAddProductCommand addProduct, IGetProductCommand getProduct,  
IEditProductCommand editProduct, IDeleteProductCommand deleteProduct)  
        {  
            this.context = context;  
            _addProduct = addProduct;  
            _getProducts = getProducts;  
            _getProduct = getProduct;  
            _editProduct = editProduct;  
            _deleteProduct = deleteProduct;  
        }  
    }  
}
```

```

// GET api/categories
[HttpGet]
public IActionResult Get([FromQuery]ProductSearch query)
    => Ok(_getProducts.Execute(query));

// GET api/categories/5
[HttpGet("{id}")]
public IActionResult Get(int id)
{
    try
    {
        var productDTO = _getProduct.Execute(id);
        return Ok(productDTO);
    }
    catch (EntityNotFoundException)
    {
        return NotFound();
    }
}

// POST api/values
[HttpPost]
public IActionResult Post([FromBody] ProductDTO dto)
{
    var produkt = new Product
    {
        ProductName = dto.ProductName
    };

    context.Product.Add(produkt);

    try
    {
        context.SaveChanges();

        return Created("/api/products/" + produkt.Id, new ProductDTO
        {
            Id = produkt.Id,
            ProductName = produkt.ProductName
        });
    }
    catch
    {
        return StatusCode(500, "An error has occurred.");
    }
}

```

```
}
```

```
// PUT api/values/5
```

```
[HttpPut("{id}")]
```

```
public IActionResult Put(int id, [FromBody] ProductDTO dto)
```

```
{
```

```
    var product = context.Product.Find(id);
```

```
    if (product == null)
```

```
    {
```

```
        return NotFound();
```

```
    }
```

```
    if (product.IsDeleted)
```

```
    {
```

```
        return NotFound();
```

```
    }
```

```
    if (context.Product.Any(c => c.ProductName == dto.ProductName))
```

```
    {
```

```
        return Conflict("Produkt sa tim imenom vec postoji.");
```

```
    }
```

```
    product.ProductName = dto.ProductName;
```

```
    try
```

```
    {
```

```
        context.SaveChanges();
```

```
        return NoContent();
```

```
    }
```

```
    catch (Exception)
```

```
    {
```

```
        return StatusCode(500, "An error has occurred.");
```

```
    }
```

```
}
```

```
// DELETE api/categories/5
```

```
[HttpDelete("{id}")]
```

```
public IActionResult Delete(int id)
```

```
{
```

```
    var product = context.Product.Find(id);
```

```
    if (product == null)
```

```
    {
```

```

        return NotFound();
    }

    if (product.IsDeleted)
        return Conflict("Produkt je vec obrisan.");

    product.IsDeleted = true;

    try
    {
        context.SaveChanges();
        return NoContent();
    }
    catch (Exception)
    {
        return StatusCode(500);
    }
}
}

```

## SupplierController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Application.Commands.Supplier;
using Application.DTO;
using Application.Exceptions;
using Application.Searches;
using EfDataAccess;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

namespace API.Controllers
{
    public class SupplierController : Controller
    {
        private readonly ProjContext context;
    }
}

```

```
private readonly IAddSupplierCommand addSupplier;
private readonly IGetSupplierCommand getSupplier;
private readonly IEditSupplierCommand editSupplier;
private readonly IDeleteSupplierCommand deleteSupplier;
private readonly IGetSuppliersCommand getSuppliers;
```

```
public SupplierController(ProjContext context, IAddSupplierCommand
addSupplier, IGetSupplierCommand getSupplier, IEditSupplierCommand
editSupplier, IDeleteSupplierCommand deleteSupplier, IGetSuppliersCommand
getSuppliers)
```

```
{
    this.context = context;
    this.addSupplier = addSupplier;
    this.getSupplier = getSupplier;
    this.editSupplier = editSupplier;
    this.deleteSupplier = deleteSupplier;
    this.getSuppliers = getSuppliers;
}
```

```
// GET: Order
```

```
public IActionResult Get([FromQuery]SupplierSearch query)
    => Ok(getSuppliers.Execute(query));
```

```
// GET: Order/Details/5
```

```
public IActionResult Get(int id)
{
    try
    {
        var supplierDTO = getSupplier.Execute(id);
        return Ok(supplierDTO);
    }
    catch (EntityNotFoundException)
    {
        return NotFound();
    }
}
```

```
[HttpPost]
```

```
public IActionResult Post([FromBody] SupplierDTO dto)
{
    var supplier = new Supplier
    {
        CompanyName = dto.CompanyName
    };
}
```

```

context.Supplier.Add(supplier);

try
{
    context.SaveChanges();

    return Created("/api/users/" + supplier.Id, new SupplierDTO
    {
        Id = supplier.Id,
        CompanyName = supplier.CompanyName,
        ContactName = supplier.ContactName
    });
}
catch
{
    return StatusCode(500, "An error has occurred.");
}
}

// PUT api/values/5
[HttpPut("{id}")]
public IActionResult Put(int id, [FromBody] SupplierDTO dto)
{
    var supplier = context.Supplier.Find(id);

    if (supplier == null)
    {
        return NotFound();
    }

    /// if (customer.IsDeleted)
    /// {
    ///     return NotFound();
    /// }

    if (context.Supplier.Any(c => c.Id == dto.Id))
    {
        return Conflict("Takav customer vec postoji.");
    }

    supplier.Id = dto.Id;

    try
    {
        context.SaveChanges();
    }
}

```

```

        return NoContent();
    }
    catch (Exception)
    {

        return StatusCode(500, "An error has occurred.");
    }
}

// DELETE api/categories/5
[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    var supplier = context.Supplier.Find(id);

    if (supplier == null)
    {
        return NotFound();
    }

    /// if (user.IsDeleted)
    ///     return Conflict("Customer je obrisan.");

    ///user.IsDeleted = true;

    try
    {
        context.SaveChanges();
        return NoContent();
    }
    catch (Exception)
    {

        return StatusCode(500);
    }
}
}
}

```

## **UserController.cs**

```

using System;
using System.Collections.Generic;

```



```

using System.Linq;
using System.Threading.Tasks;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using Application.Searches;
using EfDataAccess;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

namespace API.Controllers
{
    public class UserController : Controller
    {
        private readonly ProjContext context;
        private readonly IAddUserCommand addUser;
        private readonly IGetUserCommand getUser;
        private readonly IEditUserCommand editUser;
        private readonly IDeleteUserCommand deleteUser;
        private readonly IGetUsersCommand getUsers;

        public UserController(IGetUsersCommand users, ProjContext context,
            IAddUserCommand addUser, IGetUserCommand getUser, IEditUserCommand
            editUser, IDeleteUserCommand deleteUser)
        {
            this.context = context;
            this.addUser = addUser;
            this.getUser = getUser;
            this.editUser = editUser;
            this.deleteUser = deleteUser;
            getUsers = getUsers;
        }

        // GET api/categories
        [HttpGet]
        public IActionResult Get([FromQuery]UserSearch query)
            => Ok(getUsers.Execute(query));

        // GET api/categories/5
        [HttpGet("{id}")]
        public IActionResult Get(int id)
        {

```

```

try
{
    var userDTO = getUser.Execute(id);
    return Ok(userDTO);
}
catch (EntityNotFoundException)
{
    return NotFound();
}
}

// POST api/values
[HttpPost]
public IActionResult Post([FromBody] UserDTO dto)
{
    var user = new Users
    {
        Username = dto.Username
    };

    context.Users.Add(user);

    try
    {
        context.SaveChanges();

        return Created("/api/users/" + user.Userid, new UserDTO
        {
            Userid = user.Userid,
            Username = user.Username,
            Email=user.Email
        });
    }
    catch
    {
        return StatusCode(500, "An error has occurred.");
    }
}

// PUT api/values/5
[HttpPut("{id}")]
public IActionResult Put(int id, [FromBody] UserDTO dto)
{
    var user = context.Users.Find(id);

```

```

    if (user == null)
    {
        return NotFound();
    }

    if (user.IsDeleted)
    {
        return NotFound();
    }

    if (context.Users.Any(c => c.Username == dto.Username))
    {
        return Conflict("Username sa tim imenom vec postoji.");
    }

    user.Username = dto.Username;

    try
    {
        context.SaveChanges();
        return NoContent();
    }
    catch (Exception)
    {
        return StatusCode(500, "An error has occurred.");
    }
}

```

```

// DELETE api/categories/5
[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    var user = context.Users.Find(id);

    if (user == null)
    {
        return NotFound();
    }

    if (user.IsDeleted)
        return Conflict("Produkt je vec obrisan.");

    ///user.IsDeleted = true;

```

```

        try
        {
            context.SaveChanges();
            return NoContent();
        }
        catch (Exception)
        {
            return StatusCode(500);
        }
    }
}
}

```

## ValuesController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;

namespace API.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ValuesController : ControllerBase
    {
        // GET api/values
        [HttpGet]
        public ActionResult<IEnumerable<string>> Get()
        {
            return new string[] { "value1", "value2" };
        }

        // GET api/values/5
        [HttpGet("{id}")]
        public ActionResult<string> Get(int id)
        {
            return "value";
        }

        // POST api/values
    }
}

```

```

[HttpPost]
public void Post([FromBody] string value)
{
}

// PUT api/values/5
[HttpPut("{id}")]
public void Put(int id, [FromBody] string value)
{
}

// DELETE api/values/5
[HttpDelete("{id}")]
public void Delete(int id)
{
}
}
}

```

## Startup.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Application.Commands;
using Application.Commands.User;
using Application.Interface;
using EfCommands;
using EfCommands.UserEF;
using EfDataAccess;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Options;

namespace API
{
    public class Startup
    {
        public Startup(IConfiguration configuration)

```

```

    {
        Configuration = configuration;
    }

    public IConfiguration Configuration { get; }

    // This method gets called by the runtime. Use this method to add services to the
    container.
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
        services.AddDbContext<ProjContext>();
        services.AddTransient<IAddProductCommand, EfAddProductCommand>();
        /// services.AddTransient<IGetProductsCommand,
EfGetProductsCommand>();
        services.AddTransient<IGetProductCommand, EfGetProducttCommand>();
        services.AddTransient<IEditProductCommand, EfEditProductCommand>();
        services.AddTransient<IDeleteProductCommand,
EfDeleteProductCommand>();

        /// services.AddTransient<IGetProductsCommand,
EfGetProductsCommand>();
        services.AddTransient<IGetUserCommand, EfGetUserCommand>();
        services.AddTransient<IEditUserCommand, EfEditUserCommand>();
        services.AddTransient<IDeleteUserCommand, EfDeleteUserCommand>();
    }

    private static void NewMethod(IServiceCollection services)
    {
        services.AddTransient<IAddUserCommand, EfAddUserCommand>();
    }

    // This method gets called by the runtime. Use this method to configure the
    HTTP request pipeline.
    public void Configure(IApplicationBuilder app, IHostingEnvironment env)
    {
        if (env.IsDevelopment())
        {
            app.UseDeveloperExceptionPage();
        }
        else
        {
            app.UseHsts();
        }
    }

```

```
        app.UseHttpsRedirection();
        app.UseMvc();
    }
}
```

## **Web projekat**

### **HomeController.cs**

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Web.Models;

namespace Web.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }

        public IActionResult About()
        {
            ViewData["Message"] = "Your application description page.";

            return View();
        }

        public IActionResult Contact()
        {
            ViewData["Message"] = "Your contact page.";

            return View();
        }

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

```

    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None,
NoStore = true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
    }
}
}

```

## **ProductController.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Application.Commands;
using Application.DTO;
using Application.Exceptions;
using Application.Interface;
using Application.Searches;
using EfDataAccess;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Web.Models;

namespace Web.Controllers
{
    public class ProductController : Controller
    {
        private readonly ProjContext context;
        private readonly IAddProductCommand _addProduct;
        /// private readonly IGetProductsCommand _getProducts;
        private readonly IGetProductCommand _getProduct;
        private readonly IEditProductCommand _editProduct;
        private readonly IDeleteProductCommand _deleteProduct;

        public ProductController(ProjContext context, IAddProductCommand
addProduct, IGetProductCommand getProduct, IEditProductCommand editProduct,
IDeleteProductCommand deleteProduct)
        {
            this.context = context;

```



```
    _addProduct = addProduct;  
    /// _getProducts = getProducts;  
    _getProduct = getProduct;  
    _editProduct = editProduct;  
    _deleteProduct = deleteProduct;  
}
```

```
// GET: Product  
public ActionResult Index(ProductSearch search)  
{  
  
    return View();  
}
```

```
// GET: Product/Details/5  
public ActionResult Details(int id)  
{  
    try  
    {  
        var dto = _getProduct.Execute(id);  
        return View(dto);  
    }  
    catch (Exception)  
    {  
        return View();  
    }  
}
```

```
// GET: Product/Create  
public ActionResult Create()  
{  
    return View();  
}
```

```
// POST: Product/Create  
[HttpPost]  
[ValidateAntiForgeryToken]  
public ActionResult Create(ProductDTO dto)  
{
```

```

    if (!ModelState.IsValid) {
        return View(dto);
    }
    try
    {
        // TODO: Add insert logic here
        _addProduct.Execute(dto);
        return RedirectToAction(nameof(Index));
    }
    catch (EntityAlreadyExistsException)
    {
        TempData["error"] = "Produkt vec postoji.";
    }
    catch (Exception) {
        TempData["error"] = "Doslo je do greske.";
    }
    return View();
}

```

```

// GET: Product/Edit/5
public ActionResult Edit(int id)
{
    try
    {
        var dto = _getProduct.Execute(id);
        return View(dto);
    }
    catch (Exception)
    {

        return RedirectToAction("index");
    }

}

```

```

// POST: Product/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit(int id,[FromBody] ProductDTO dto)
{
    if (!ModelState.IsValid) {
        return View(dto);
    }
    try

```

```

    {
        // TODO: Add update logic here
        _editProduct.Execute(dto);
        return RedirectToAction(nameof(Index));
    }
    catch (EntityNotFoundException)
    {
        return RedirectToAction(nameof(Index));
    }
    catch (EntityAlreadyExistsException) {
        TempData["error"] = "Produkt sa tim imenom postoji";
        return View(dto);
    }
}

```

// GET: Product/Delete/5

// POST: Product/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id)

```

{
    var product = context.Product.Find(id);

    if (product == null)
    {
        return NotFound();
    }

    if (product.IsDeleted)
        return Conflict("That product is already deleted.");

    product.IsDeleted = true;

    try
    {
        context.SaveChanges();
        return NoContent();
    }
    catch (Exception)
    {
        return StatusCode(500);
    }
}

```

```

    }
}

public IActionResult Product() {
    var vm = new ProductViewModel();
    var products = new List<ProductDTO>();
    vm.Products = products;
    return View(products);
}

}

}

```

## **UserController.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Application.Commands.User;
using Application.DTO;
using Application.Exceptions;
using EfDataAccess;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Web.Models;

namespace Web.Controllers
{
    public class UserController : Controller
    {
        private readonly ProjContext context;
        private readonly IAddUserCommand addUser;
        private readonly IGetUserCommand getUser;
        private readonly IEditUserCommand editUser;
        private readonly IDeleteUserCommand deleteUser;
        private readonly IGetUsersCommand getUsers;

        public UserController(ProjContext context, IAddUserCommand addUser,
            IGetUserCommand getUser, IEditUserCommand editUser, IDeleteUserCommand
            deleteUser, IGetUsersCommand getUsers)
        {
            this.context = context;

```

```
this.addUser = addUser;
this.getUser = getUser;
this.editUser = editUser;
this.deleteUser = deleteUser;
this.getUsers = getUsers;
}
```

```
// GET: User
public ActionResult Index()
{
    return View();
}
```

```
// GET: User/Details/5
public ActionResult Details(int id)
{
    try
    {
        var dto = getUser.Execute(id);
        return View(dto);
    }
    catch (Exception)
    {
        return View();
    }
}
```

```
// GET: User/Create
public ActionResult Create()
{
    return View();
}
```

```
// POST: User/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(UserDTO dto)
{
    if (!ModelState.IsValid)
    {
```

```

        return View(dto);
    }
    try
    {
        // TODO: Add insert logic here
        addUser.Execute(dto);
        return RedirectToAction(nameof(Index));
    }
    catch (EntityAlreadyExistsException)
    {
        TempData["error"] = "Korisnik vec postoji.";
    }
    catch (Exception)
    {
        TempData["error"] = "Doslo je do greske.";
    }
    return View();
}

```

```

// GET: User/Edit/5
public ActionResult Edit(int id)
{
    try
    {
        var dto = getUser.Execute(id);
        return View(dto);
    }
    catch (Exception)
    {

        return RedirectToAction("index");
    }
}

```

```

// POST: User/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit(int id, [FromBody] UserDTO dto)
{
    if (!ModelState.IsValid)
    {
        return View(dto);
    }
}

```

```

try
{
    // TODO: Add update logic here
    editUser.Execute(dto);
    return RedirectToAction(nameof(Index));
}
catch (EntityNotFoundException)
{
    return RedirectToAction(nameof(Index));
}
catch (EntityAlreadyExistsException)
{
    TempData["error"] = "Korisnik sa tim imenom postoji";
    return View(dto);
}
}

```

// GET: User/Delete/5

// POST: User/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id)

```

{
    var user = context.Users.Find(id);

    if (user == null)
    {
        return NotFound();
    }

    if (user.IsDeleted)
        return Conflict("Korisnik je vec obrisan.");

```

////user.IsDeleted = true;

```

try
{
    context.SaveChanges();
    return NoContent();
}
catch (Exception)
{

```

```

        return StatusCode(500);
    }
}

public IActionResult User()
{
    /// var vm = new UserViewModel();
    // var products = new List<ProductDTO>();
    // vm.Products = products;
    return View();
}

}
}

```

## Startup.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Application.Commands;
using Application.Commands.User;
using Application.Interface;
using EfCommands;
using EfCommands.User;
using EfCommands.UserEF;
using EfDataAccess;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;

namespace Web
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }
    }
}

```



```
}
```

```
public IConfiguration Configuration { get; }
```

// This method gets called by the runtime. Use this method to add services to the container.

```
public void ConfigureServices(IServiceCollection services)
```

```
{
```

```
    services.Configure<CookiePolicyOptions>(options =>
```

```
    {
```

// This lambda determines whether user consent for non-essential cookies is needed for a given request.

```
        options.CheckConsentNeeded = context => true;
```

```
        options.MinimumSameSitePolicy = SameSiteMode.None;
```

```
    });
```

```
services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
```

```
services.AddDbContext<ProjContext>();
```

```
services.AddTransient<IAddProductCommand, EfAddProductCommand>();
```

```
/// services.AddTransient<IGetProductsCommand,  
EfGetProductsCommand>();
```

```
services.AddTransient<IGetProductCommand, EfGetProducttCommand>();
```

```
services.AddTransient<IEditProductCommand, EfEditProductCommand>();
```

```
services.AddTransient<IDeleteProductCommand,  
EfDeleteProductCommand>();
```

```
services.AddTransient<IAddUserCommand, EfAddUserCommand>();
```

```
/// services.AddTransient<IGetProductsCommand,  
EfGetProductsCommand>();
```

```
services.AddTransient<IGetUserCommand,  
EfCommands.UserEF.EfGetUserCommand>();
```

```
services.AddTransient<IEditUserCommand, EfEditUserCommand>();
```

```
services.AddTransient<IDeleteUserCommand, EfDeleteUserCommand>();
```

```
}
```

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
```

```
{
```

```
    if (env.IsDevelopment())
```

```
    {
```

```
        app.UseDeveloperExceptionPage();
```

```
    }
```

```

else
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseCookiePolicy();

app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller=Home}/{action=Index}/{id?}");
    });
}
}
}

```

## Views folder

### Product subfolder

#### Create.cshtml

@model Application.DTO.ProductDTO

```

@{
    ViewData["Title"] = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

<h2>Create</h2>

<h4>ProductDTO</h4>

```

@if (TempData["error"] != null) {
    <p class="text-danger">@TempData["error"]</p>
}

```

<hr />

<div class="row">

<div class="col-md-4">

```

<form asp-action="Create" method="post">
  <div asp-validation-summary="ModelOnly" class="text-danger"></div>

  <div class="form-group">
    <label asp-for="ProductName" class="control-label"></label>
    <input asp-for="ProductName" class="form-control" />
    <span asp-validation-for="ProductName" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="SupplierId" class="control-label"></label>
    <input asp-for="SupplierId" class="form-control" />
    <span asp-validation-for="SupplierId" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="UnitPrice" class="control-label"></label>
    <input asp-for="UnitPrice" class="form-control" />
    <span asp-validation-for="UnitPrice" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="Package" class="control-label"></label>
    <input asp-for="Package" class="form-control" />
    <span asp-validation-for="Package" class="text-danger"></span>
  </div>
  <div class="form-group">
    <div class="checkbox">
      <label>
        <input asp-for="IsDiscontinued" /> @Html.DisplayNameFor(model
=> model.IsDiscontinued)
      </label>
    </div>
  </div>
  <div class="form-group">
    <input type="submit" value="Create" class="btn btn-default" />
  </div>
</form>
</div>
</div>

<div>
  <a asp-action="Index">Back to List</a>
</div>

```

## Delete.cshtml

@model Application.DTO.ProductDTO

```
@{
    ViewData["Title"] = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>

<div>

<h4>ProductDTO</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.Id)

</dt>

<dd>

@Html.DisplayFor(model => model.Id)

</dd>

<dt>

@Html.DisplayNameFor(model => model.ProductName)

</dt>

<dd>

@Html.DisplayFor(model => model.ProductName)

</dd>

<dt>

@Html.DisplayNameFor(model => model.SupplierId)

</dt>

<dd>

@Html.DisplayFor(model => model.SupplierId)

</dd>

<dt>

@Html.DisplayNameFor(model => model.UnitPrice)

</dt>

<dd>

@Html.DisplayFor(model => model.UnitPrice)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Package)

</dt>

<dd>

```

        @Html.DisplayFor(model => model.Package)
    </dd>
    <dt>
        @Html.DisplayNameFor(model => model.IsDiscontinued)
    </dt>
    <dd>
        @Html.DisplayFor(model => model.IsDiscontinued)
    </dd>
</dl>

<form asp-action="Delete">
    <input type="submit" value="Delete" class="btn btn-default" /> |
    <a asp-action="Index">Back to List</a>
</form>
</div>

```

## Details.cshtml

```

@model Application.DTO.ProductDTO

@{
    ViewData["Title"] = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Details</h2>
@if (Model == null)
{
    <h1>Produkt ne postoji</h1>
}
else
{
    <div>
        <h4>ProductDTO</h4>
        <hr />
        <dl class="dl-horizontal">
            <dt>
                @Html.DisplayNameFor(model => model.Id)
            </dt>
            <dd>
                @Html.DisplayFor(model => model.Id)
            </dd>

```

```

<dt>
    @Html.DisplayNameFor(model => model.ProductName)
</dt>
<dd>
    @Html.DisplayFor(model => model.ProductName)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.SupplierId)
</dt>
<dd>
    @Html.DisplayFor(model => model.SupplierId)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.UnitPrice)
</dt>
<dd>
    @Html.DisplayFor(model => model.UnitPrice)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.Package)
</dt>
<dd>
    @Html.DisplayFor(model => model.Package)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.IsDiscontinued)
</dt>
<dd>
    @Html.DisplayFor(model => model.IsDiscontinued)
</dd>
</dl>
</div>

<div>
    @Html.ActionLink("Edit", "Edit", new { /* id = Model.PrimaryKey */ }) |
    <a asp-action="Index">Back to List</a>
</div>
}

```

## Index.cshtml

```
@model IEnumerable<Application.DTO.ProductDTO>
```

```
@{  
    ViewData["Title"] = "Index";  
    Layout = "~/Views/Shared/_Layout.cshtml";  
}
```

```
<h2>Index</h2>
```

```
<p>  
    <a asp-action="Create">Create New</a>
```

```
</p>
```

```
<table class="table">  
    <thead>  
        <tr>  
            <th>  
                @Html.DisplayNameFor(model => model.Id)  
            </th>  
            <th>  
                @Html.DisplayNameFor(model => model.ProductName)  
            </th>  
            <th>  
                @Html.DisplayNameFor(model => model.SupplierId)  
            </th>  
            <th>  
                @Html.DisplayNameFor(model => model.UnitPrice)  
            </th>  
            <th>  
                @Html.DisplayNameFor(model => model.Package)  
            </th>  
            <th>  
                @Html.DisplayNameFor(model => model.IsDiscontinued)  
            </th>  
            <th></th>  
        </tr>  
    </thead>  
    <tbody>  
        @if (Model != null)  
        {  
            @foreach (var item in Model)
```

```

<tr>
  <td>
    @Html.DisplayFor(modelItem => item.Id)
  </td>
  <td>
    @Html.DisplayFor(modelItem => item.ProductName)
  </td>
  <td>
    @Html.DisplayFor(modelItem => item.SupplierId)
  </td>
  <td>
    @Html.DisplayFor(modelItem => item.UnitPrice)
  </td>
  <td>
    @Html.DisplayFor(modelItem => item.Package)
  </td>
  <td>
    @Html.DisplayFor(modelItem => item.IsDiscontinued)
  </td>
  <td>
    @Html.ActionLink("Edit", "Edit", new { /* id=item.PrimaryKey */ })
|
    @Html.ActionLink("Details", "Details", new { /* id=item.PrimaryKey
*/ }) |
    @Html.ActionLink("Delete", "Delete", new { /* id=item.PrimaryKey
*/ })
  </td>
</tr>
}}
</tbody>
</table>

```

## Edit.cshtml

```
@model Application.DTO.ProductDTO
```

```
@{
    ViewData["Title"] = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

```
<h2>Edit</h2>
```

```
<h4>ProductDTO</h4>
```



```

<hr />
<div class="row">
  <div class="col-md-4">
    <form asp-action="Edit">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>

      <div class="form-group">
        <label asp-for="ProductName" class="control-label"></label>
        <input asp-for="ProductName" class="form-control" />
        <span asp-validation-for="ProductName" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="SupplierId" class="control-label"></label>
        <input asp-for="SupplierId" class="form-control" />
        <span asp-validation-for="SupplierId" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="UnitPrice" class="control-label"></label>
        <input asp-for="UnitPrice" class="form-control" />
        <span asp-validation-for="UnitPrice" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Package" class="control-label"></label>
        <input asp-for="Package" class="form-control" />
        <span asp-validation-for="Package" class="text-danger"></span>
      </div>
      <div class="form-group">
        <div class="checkbox">
          <label>
            <input asp-for="IsDiscontinued" /> @Html.DisplayNameFor(model
=> model.IsDiscontinued)
          </label>
        </div>
      </div>
      <div class="form-group">
        <input type="submit" value="Save" class="btn btn-default" />
      </div>
    </form>
  </div>
</div>

<div>
  <a asp-action="Index">Back to List</a>
</div>

```

```
@model Web.Models.ProductViewModel
```

## Product

<html class="no-js" lang="zxx">

```
<meta charset="utf-8">
```

```
<meta http-equiv="x-ua-compatible" content="ie=edge">
```

&lt;title&gt;Oswan - eCommerce HTML5 Template&lt;/title&gt;

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

&lt;link rel="shortcut icon" type="image/x-icon" href="assets/img/favicon.png"&gt;

&lt;link rel="stylesheet" href="assets/css/bootstrap.min.css"&gt;

&lt;link rel="stylesheet" href="assets/css/animate.css"&gt;

&lt;link rel="stylesheet" href="assets/css/owl.carousel.min.css"&gt;

```
<link rel="stylesheet" href="assets/css/chosen.min.css">
```

```
<link rel="stylesheet" href="assets/css/icofont.css">
```

```
<link rel="stylesheet" href="assets/css/themify-icons.css">
```

```
<link rel="stylesheet" href="assets/css/font-awesome.min.css">
```

&lt;link rel="stylesheet" href="assets/css/meanmenu.min.css"&gt;

&lt;link rel="stylesheet" href="assets/css/bundle.css"&gt;

```
<link rel="stylesheet" href="assets/css/style.css">
```

```
<link rel="stylesheet" href="assets/css/responsive.css">
```

```
<script src="assets/js/vendor/modernizr-2.8.3.min.js"></script>
```

&lt;body&gt;

<div class="wrapper">

```
<!-- header start -->
```

&lt;header&gt;

<div class="header-area transparent-bar ptb-55">

<div class="container">

<div class="col-lg-4 col-md-4 col-4">

<div class="logo-small-device">

```

        <a href="index.html"></a>
    </div>
</div>
<div class="col-lg-8 col-md-8 col-8">
    <div class="header-contact-menu-wrapper pl-45">
        <div class="header-contact">
            <p>WANT TO TALK WITH US +01254 265 987</p>
        </div>
        <div class="menu-wrapper text-center">
            <button class="menu-toggle">
                
                
            </button>
            <div class="main-menu">
                <nav>
                    <ul>
                        <li><a href="index.html">home</a></li>
                        <li class="active"><a href="about-us.html">about us
</a></li>
                        <li>
                            <a href="#">shop</a>
                            <ul>
                                <li><a href="shop.html">shop</a></li>
                                <li><a href="product-details.html">product
details</a></li>
                                <li><a href="checkout.html">checkout</a></li>
                                <li><a href="wishlist.html">wishlist</a></li>
                                <li><a href="cart.html">cart</a></li>
                            </ul>
                        </li>
                        <li>
                            <a href="#">pages</a>
                            <ul>
                                <li><a href="about-us.html">about us</a></li>
                                <li><a href="cart.html">cart page</a></li>
                                <li><a href="checkout.html">checkout</a></li>
                                <li><a href="wishlist.html">wishlist</a></li>
                                <li><a href="login-register.html">login</a></li>
                                <li><a href="contact.html">contact</a></li>
                            </ul>
                        </li>
                        <li>

```

```

        <a href="#">blog</a>
        <ul>
            <li><a href="blog.html">blog</a></li>
            <li><a href="blog-details.html">blog
details</a></li>
            </ul>
        </li>
        <li><a href="contact.html">contact us</a></li>
        </ul>
    </nav>
</div>
</div>
</div>
<div class="header-cart cart-small-device">
    <button class="icon-cart">
        <i class="ti-shopping-cart"></i>
        <span class="count-style">02</span>
        <span class="count-price-add">$295.95</span>
    </button>
    <div class="shopping-cart-content">
        <ul>
            <li class="single-shopping-cart">
                <div class="shopping-cart-img">
                    <a href="#"></a>
                </div>
                <div class="shopping-cart-title">
                    <h3><a href="#">Gloriori GSX 250 R </a></h3>
                    <span>Price: $275</span>
                    <span>Qty: 01</span>
                </div>
                <div class="shopping-cart-delete">
                    <a href="#"><i class="icofont icofont-ui-
delete"></i></a>
                </div>
            </li>
            <li class="single-shopping-cart">
                <div class="shopping-cart-img">
                    <a href="#"></a>
                </div>
                <div class="shopping-cart-title">
                    <h3><a href="#">Demonissi Gori</a></h3>
                    <span>Price: $275</span>
                    <span class="qty">Qty: 01</span>

```

```

        </div>
        <div class="shopping-cart-delete">
            <a href="#"><i class="icofont icofont-ui-
delete"></i></a>
        </div>
    </li>
    <li class="single-shopping-cart">
        <div class="shopping-cart-img">
            <a href="#"></a>
        </div>
        <div class="shopping-cart-title">
            <h3><a href="#">Demonissi Gori</a></h3>
            <span>Price: $275</span>
            <span class="qty">Qty: 01</span>
        </div>
        <div class="shopping-cart-delete">
            <a href="#"><i class="icofont icofont-ui-
delete"></i></a>
        </div>
    </li>
</ul>
<div class="shopping-cart-total">
    <h4>total: <span>$550.00</span></h4>
</div>
<div class="shopping-cart-btn">
    <a class="btn-style cr-btn" href="#">checkout</a>
</div>
</div>
</div>
<div class="mobile-menu-area col-12">
    <div class="mobile-menu">
        <nav id="mobile-menu-active">
            <ul class="menu-overflow">
                <li><a href="index.html">HOME</a></li>
                <li>
                    <a href="#">pages</a>
                    <ul>
                        <li><a href="about-us.html">about us</a></li>
                        <li><a href="cart.html">cart page</a></li>
                        <li><a href="checkout.html">checkout</a></li>
                        <li><a href="wishlist.html">wishlist</a></li>
                        <li><a href="login-register.html">login</a></li>
                        <li><a href="contact.html">contact</a></li>
                    </ul>
                </li>
            </ul>
        </nav>
    </div>
</div>

```

```

        </ul>
      </li>
      <li>
        <a href="#">shop</a>
        <ul>
          <li><a href="shop.html">shop</a></li>
          <li><a href="product-details.html">product
details</a></li>
          <li><a href="checkout.html">checkout</a></li>
          <li><a href="wishlist.html">wishlist</a></li>
          <li><a href="cart.html">cart</a></li>
        </ul>
      </li>
      <li>
        <a href="#">BLOG</a>
        <ul>
          <li><a href="blog.html">blog page</a></li>
          <li><a href="blog-details.html">blog details</a></li>
        </ul>
      </li>
      <li><a href="contact.html"> Contact us</a></li>
    </ul>
  </nav>
</div>
</div>
</div>
</div>
<div class="header-cart-wrapper">
  <div class="header-cart">
    <button class="icon-cart">
      <i class="ti-shopping-cart"></i>
      <span class="count-style">02</span>
      <span class="count-price-add">$295.95</span>
    </button>
    <div class="shopping-cart-content">
      <ul>
        <li class="single-shopping-cart">
          <div class="shopping-cart-img">
            <a href="#"></a>
          </div>
          <div class="shopping-cart-title">
            <h3><a href="#">Gloriori GSX 250 R </a></h3>
            <span>Price: $275</span>
            <span>Qty: 01</span>
          </div>
        </li>
      </ul>
    </div>
  </div>
</div>

```

```

        </div>
        <div class="shopping-cart-delete">
            <a href="#"><i class="icofont icofont-ui-delete"></i></a>
        </div>
    </li>
    <li class="single-shopping-cart">
        <div class="shopping-cart-img">
            <a href="#"></a>

        </div>
        <div class="shopping-cart-title">
            <h3><a href="#">Demonissi Gori</a></h3>
            <span>Price: $275</span>
            <span class="qty">Qty: 01</span>
        </div>
        <div class="shopping-cart-delete">
            <a href="#"><i class="icofont icofont-ui-delete"></i></a>
        </div>
    </li>
    <li class="single-shopping-cart">
        <div class="shopping-cart-img">
            <a href="#"></a>

        </div>
        <div class="shopping-cart-title">
            <h3><a href="#">Demonissi Gori</a></h3>
            <span>Price: $275</span>
            <span class="qty">Qty: 01</span>
        </div>
        <div class="shopping-cart-delete">
            <a href="#"><i class="icofont icofont-ui-delete"></i></a>
        </div>
    </li>
</ul>
<div class="shopping-cart-total">
    <h4>total: <span>$550.00</span></h4>
</div>
<div class="shopping-cart-btn">
    <a class="btn-style cr-btn" href="#">checkout</a>
</div>
</div>
</div>
</div>
</div>
</header>

```

```

<div class="slider-area">
  <div class="slider-active owl-carousel">
    <div class="single-slider slider-1" style="background-image:
url(assets/img/slider/slider-bg.jpg)">
      <div class="container">
        <div class="slider-content slider-animated-1">
          <div class="slider-img text-center">
            
          </div>
          <div class="slider-text-img">
            <h6 class="animated">BOOK YOUR BIKE INSTANTLY AND
ENJOY DISCOUNT</h6>
            
          </div>
            <h2 class="animated">MOTORCYCLE</h2>
          </div>
        </div>
      <div class="single-slider slider-1" style="background-image:
url(assets/img/slider/slider-bg.jpg)">
        <div class="container">
          <div class="slider-content slider-animated-1">
            <div class="slider-img text-center">
              
            </div>
            <div class="slider-text-img">
              <h6 class="animated">BOOK YOUR BIKE INSTANTLY AND
ENJOY DISCOUNT</h6>
              
            </div>
              <h2 class="animated">MOTORCYCLE</h2>
            </div>
          </div>
        <div class="single-slider slider-1" style="background-image:
url(assets/img/slider/slider-bg.jpg)">
          <div class="container">
            <div class="slider-content slider-animated-1">
              <div class="slider-img text-center">
                

```



```
</div>
<div class="slider-text-img">
  <h6 class="animated">BOOK YOUR BIKE INSTANTLY AND
ENJOY DISCOUNT</h6>
  
</div>
<h2 class="animated">MOTORCYCLE</h2>
</div>
</div>
</div>
<div class="slider-social">
  <ul>
    <li class="facebook"><a href="#"><i class="icofont icofont-social-
facebook"></i></a></li>
    <li class="twitter"><a href="#"><i class="icofont icofont-social-
twitter"></i></a></li>
    <li class="pinterest"><a href="#"><i class="icofont icofont-social-
pinterest"></i></a></li>
  </ul>
</div>
<div class="language-currency-wrapper">
  <div class="language-currency">
    <div class="language">
      <select class="select-header orderby">
        <option value="">ENG</option>
        <option value="">BANGLA </option>
        <option value="">HINDI</option>
      </select>
    </div>
    <div class="currency">
      <select class="select-header orderby">
        <option value="">$USD</option>
        <option value="">US </option>
        <option value="">EURO</option>
      </select>
    </div>
  </div>
</div>
</div>
<div class="overview-area pt-135">
  <div class="container">
    <div class="row">
      <div class="col-lg-6 col-md-12">
```

```

<div class="overview-content">
  <h1><span>OSWAN</span> WORLD MOST </h1>
  <h2>LATGEST <span>MOTORCYCLE STORE</span></h2>
  <p><span>OSWAN</span> the most latgest bike store in the wold
can serve you latest      qulity of motorcycle also you can sell here your
motorcycle it quo minus iduod maxie placeat facere possimus, omnis voluptas
assumenda est, omnis dolor llendus. Temporibus autem quibusdam </p>
  <div class="question-area">
    <h4>HAVE ANY QUESTION? </h4>
    <div class="question-contact">
      <div class="question-icon">
        <i class="icofont icofont-phone"></i>
      </div>
      <div class="question-content-number">
        <h6> 01245 658 698</h6>
      </div>
    </div>
  </div>
</div>
</div>
</div>
<div class="col-lg-6 col-md-12">
  <div class="overview-img">
    
  </div>
</div>
</div>
</div>
<div class="banner-area pt-135 pb-120">
  <div class="container">
    <div class="row">
      <div class="col-md-4 col-lg-4">
        <div class="banner-wrapper mb-30">
          <a href="#"></a>
          <div class="banner-content">
            <h2>BUY NEW BIKE</h2>
          </div>
        </div>
      </div>
      <div class="col-md-4 col-lg-4">
        <div class="banner-wrapper mb-30">
          <a href="#"></a>
          <div class="banner-content">

```

```

        <h2>SELL YOUR BIKE</h2>
    </div>
</div>
</div>
<div class="col-md-4 col-lg-4">
    <div class="banner-wrapper mb-30">
        <a href="#"></a>
        <div class="banner-content">
            <h2>FIND SPARE PARTS</h2>
        </div>
    </div>
</div>
</div>
</div>
<div class="product-area pb-190">
    <div class="container">
        <div class="section-title text-center mb-50">
            <h2>CHOOSE YOUR BIKE</h2>
            <p><span>OSWAN</span> the most latgest bike store in the wold can
serve you latest      qulity of motorcycle also you can sell here your
motorcycle</p>
        </div>
        <div class="product-tab-list text-center mb-80 nav product-menu-mrg"
role="tablist">
            <a class="active" href="#home1" data-toggle="tab">
                <h4>NEW BIKES </h4>
            </a>
            <a href="#home2" data-toggle="tab">
                <h4> USED BIKES </h4>
            </a>
        </div>
        <div class="tab-content jump">
            <div class="tab-pane active" id="home1">
                <div class="product-slider-active owl-carousel">
                    <div class="product-wrapper-bundle">
                        <div class="product-wrapper">
                            <div class="product-img">
                                <a href="product-details.html">
                                    
                                </a>

                                @foreach (var product in Model.Products)
                                {

```

```

<div class="product-item-dec">
  <ul>
    <li>@product.ProductName</li>

  </ul>
</div>
}

```

```

<div class="product-item-dec">
  <ul>
    <li>2018</li>
    <li>MANUAL</li>
    <li>PETROL</li>
    <li>250 CC</li>
  </ul>
</div>

```

```

<div class="product-action">
  <a class="action-plus-2" title="Add To Cart" href="#">
    <i class=" ti-shopping-cart"></i>
  </a>
  <a class="action-cart-2" title="Wishlist" href="#">
    <i class=" ti-heart"></i>
  </a>
  <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
    <i class=" ti-zoom-in"></i>
  </a>
</div>
<div class="product-content-wrapper">
  <div class="product-title-spreed">
    <h4><a href="product-details.html">Gloriori GSX 250
R</a></h4>

```

```

    <span>6600 RPM</span>
  </div>
  <div class="product-price">
    <span>$2549</span>
  </div>
</div>

```

```

</div>
<div class="product-wrapper">
  <div class="product-img">
    <a href="product-details.html">
      
    </a>

```

```

<div class="product-item-dec">
  <ul>
    <li>2018</li>
    <li>MANUAL</li>
    <li>PETROL</li>
    <li>480 CC</li>
  </ul>
</div>
<div class="product-action">
  <a class="action-plus-2" title="Add To Cart" href="#">
    <i class=" ti-shopping-cart"></i>
  </a>
  <a class="action-cart-2" title="Wishlist" href="#">
    <i class=" ti-heart"></i>
  </a>
  <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
    <i class=" ti-zoom-in"></i>
  </a>
</div>
<div class="product-content-wrapper">
  <div class="product-title-spreed">
    <h4><a href="product-details.html">Demonissi
Gori</a></h4>

    <span>6700 RPM</span>
  </div>
  <div class="product-price">
    <span>$2549</span>
  </div>
</div>
</div>
</div>
<div class="product-wrapper-bundle">
  <div class="product-wrapper">
    <div class="product-img">
      <a href="product-details.html">
        
      </a>
    <div class="product-item-dec">
      <ul>
        <li>2018</li>
        <li>MANUAL</li>
        <li>PETROL</li>
        <li>200 CC</li>
      </ul>
    </div>
  </div>
</div>

```

```

        </ul>
    </div>
    <div class="product-action">
        <a class="action-plus-2" title="Add To Cart" href="#">
            <i class=" ti-shopping-cart"></i>
        </a>
        <a class="action-cart-2" title="Wishlist" href="#">
            <i class=" ti-heart"></i>
        </a>
        <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
            <i class=" ti-zoom-in"></i>
        </a>
    </div>
    <div class="product-content-wrapper">
        <div class="product-title-spreed">
            <h4><a href="product-details.html">Klager GSX 250
R</a></h4>

            <span>5500 RPM</span>
        </div>
        <div class="product-price">
            <span>$2549</span>
        </div>
    </div>
</div>
<div class="product-wrapper">
    <div class="product-img">
        <a href="product-details.html">
            
        </a>
    <div class="product-item-dec">
        <ul>
            <li>2018</li>
            <li>MANUAL</li>
            <li>PETROL</li>
            <li>150 CC</li>
        </ul>
    </div>
    <div class="product-action">
        <a class="action-plus-2" title="Add To Cart" href="#">
            <i class=" ti-shopping-cart"></i>
        </a>
        <a class="action-cart-2" title="Wishlist" href="#">
            <i class=" ti-heart"></i>

```

```

        </a>
        <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
            <i class=" ti-zoom-in"></i>
        </a>
    </div>
    <div class="product-content-wrapper">
        <div class="product-title-spreed">
            <h4><a href="product-details.html">Maxclon ZPE
54</a></h4>

            <span>3300 RPM</span>
        </div>
        <div class="product-price">
            <span>$2549</span>
        </div>
    </div>
</div>
</div>
<div class="product-wrapper-bundle">
    <div class="product-wrapper">
        <div class="product-img">
            <a href="product-details.html">
                
            </a>
        <div class="product-item-dec">
            <ul>
                <li>2018</li>
                <li>MANUAL</li>
                <li>PETROL</li>
                <li>150 CC</li>
            </ul>
        </div>
        <div class="product-action">
            <a class="action-plus-2" title="Add To Cart" href="#">
                <i class=" ti-shopping-cart"></i>
            </a>
            <a class="action-cart-2" title="Wishlist" href="#">
                <i class=" ti-heart"></i>
            </a>
            <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
                <i class=" ti-zoom-in"></i>
            </a>
        </div>
    </div>
</div>

```

Max</a></h4>

```
<div class="product-content-wrapper">
  <div class="product-title-spreed">
    <h4><a href="product-details.html">Matrio

    <span>4600 RPM</span>
  </div>
  <div class="product-price">
    <span>$2549</span>
  </div>
</div>
</div>
<div class="product-wrapper">
  <div class="product-img">
    <a href="product-details.html">
      
    </a>
  <div class="product-item-dec">
    <ul>
      <li>2018</li>
      <li>MANUAL</li>
      <li>PETROL</li>
      <li>250 CC</li>
    </ul>
  </div>
  <div class="product-action">
    <a class="action-plus-2" title="Add To Cart" href="#">
      <i class=" ti-shopping-cart"></i>
    </a>
    <a class="action-cart-2" title="Wishlist" href="#">
      <i class=" ti-heart"></i>
    </a>
    <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
      <i class=" ti-zoom-in"></i>
    </a>
  </div>
<div class="product-content-wrapper">
  <div class="product-title-spreed">
    <h4><a href="product-details.html">Rigoniss Z

    <span>8000 RPM</span>
  </div>
  <div class="product-price">
    <span>$2549</span>
```

1000</a></h4>



```

        </div>
    </div>
</div>
</div>
</div>
<div class="product-wrapper-bundle">
    <div class="product-wrapper">
        <div class="product-img">
            <a href="product-details.html">
                
            </a>
            <div class="product-item-dec">
                <ul>
                    <li>2018</li>
                    <li>MANUAL</li>
                    <li>PETROL</li>
                    <li>250 CC</li>
                </ul>
            </div>
            <div class="product-action">
                <a class="action-plus-2" title="Add To Cart" href="#">
                    <i class=" ti-shopping-cart"></i>
                </a>
                <a class="action-cart-2" title="Wishlist" href="#">
                    <i class=" ti-heart"></i>
                </a>
                <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
                    <i class=" ti-zoom-in"></i>
                </a>
            </div>
            <div class="product-content-wrapper">
                <div class="product-title-spreed">
                    <h4><a href="product-details.html">Klager GSX 250
R</a></h4>
                    <span>5500 RPM</span>
                </div>
                <div class="product-price">
                    <span>$2549</span>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="product-wrapper">
    <div class="product-img">

```

```

<a href="product-details.html">
  
</a>
<div class="product-item-dec">
  <ul>
    <li>2018</li>
    <li>MANUAL</li>
    <li>PETROL</li>
    <li>350 CC</li>
  </ul>
</div>
<div class="product-action">
  <a class="action-plus-2" title="Add To Cart" href="#">
    <i class=" ti-shopping-cart"></i>
  </a>
  <a class="action-cart-2" title="Wishlist" href="#">
    <i class=" ti-heart"></i>
  </a>
  <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
    <i class=" ti-zoom-in"></i>
  </a>
</div>
<div class="product-content-wrapper">
  <div class="product-title-spreed">
    <h4><a href="product-details.html">Maxclon ZPE
54</a></h4>

    <span>3300 RPM</span>
  </div>
  <div class="product-price">
    <span>$2549</span>
  </div>
</div>
</div>
</div>
</div>
<div class="tab-pane" id="home2">
  <div class="product-slider-active owl-carousel">
    <div class="product-wrapper-bundle">
      <div class="product-wrapper mb-30">
        <div class="product-img">
          <a href="product-details.html">
            

```

```

</a>
<div class="product-item-dec">
  <ul>
    <li>2018</li>
    <li>MANUAL</li>
    <li>PETROL</li>
    <li>250 CC</li>
  </ul>
</div>
<div class="product-action">
  <a class="action-plus-2" title="Add To Cart" href="#">
    <i class=" ti-shopping-cart"></i>
  </a>
  <a class="action-cart-2" title="Wishlist" href="#">
    <i class=" ti-heart"></i>
  </a>
  <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
    <i class=" ti-zoom-in"></i>
  </a>
</div>
<div class="product-content-wrapper">
  <div class="product-title-spreed">
    <h4><a href="product-details.html">Gloriori GSX 250
R</a></h4>
    <span>6600 RPM</span>
  </div>
  <div class="product-price">
    <span>$2549</span>
  </div>
</div>
</div>
<div class="product-wrapper mb-30">
  <div class="product-img">
    <a href="product-details.html">
      
    </a>
  <div class="product-item-dec">
    <ul>
      <li>2018</li>
      <li>MANUAL</li>
      <li>PETROL</li>
      <li>480 CC</li>
    </ul>

```

```

</div>
<div class="product-action">
  <a class="action-plus-2" title="Add To Cart" href="#">
    <i class=" ti-shopping-cart"></i>
  </a>
  <a class="action-cart-2" title="Wishlist" href="#">
    <i class=" ti-heart"></i>
  </a>
  <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
    <i class=" ti-zoom-in"></i>
  </a>
</div>
<div class="product-content-wrapper">
  <div class="product-title-spreed">
    <h4><a href="product-details.html">Demonissi
Gori</a></h4>

    <span>6700 RPM</span>
  </div>
  <div class="product-price">
    <span>$2549</span>
  </div>
</div>
</div>
</div>
<div class="product-wrapper-bundle">
  <div class="product-wrapper mb-30">
    <div class="product-img">
      <a href="product-details.html">
        
      </a>
    <div class="product-item-dec">
      <ul>
        <li>2018</li>
        <li>MANUAL</li>
        <li>PETROL</li>
        <li>200 CC</li>
      </ul>
    </div>
  </div>
  <div class="product-action">
    <a class="action-plus-2" title="Add To Cart" href="#">
      <i class=" ti-shopping-cart"></i>
    </a>
    <a class="action-cart-2" title="Wishlist" href="#">

```

```

        <i class=" ti-heart"></i>
    </a>
    <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
        <i class=" ti-zoom-in"></i>
    </a>
</div>
<div class="product-content-wrapper">
    <div class="product-title-spreed">
        <h4><a href="product-details.html">Klager GSX 250
R</a></h4>

        <span>5500 RPM</span>
    </div>
    <div class="product-price">
        <span>$2549</span>
    </div>
</div>
</div>
</div>
<div class="product-wrapper mb-30">
    <div class="product-img">
        <a href="product-details.html">
            
        </a>
    <div class="product-item-dec">
        <ul>
            <li>2018</li>
            <li>MANUAL</li>
            <li>PETROL</li>
            <li>150 CC</li>
        </ul>
    </div>
    <div class="product-action">
        <a class="action-plus-2" title="Add To Cart" href="#">
            <i class=" ti-shopping-cart"></i>
        </a>
        <a class="action-cart-2" title="Wishlist" href="#">
            <i class=" ti-heart"></i>
        </a>
        <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
            <i class=" ti-zoom-in"></i>
        </a>
    </div>
</div>
<div class="product-content-wrapper">

```

54</a></h4>

```
<div class="product-title-spreed">
  <h4><a href="product-details.html">Maxclon ZPE

  <span>3300 RPM</span>
</div>
<div class="product-price">
  <span>$2549</span>
</div>
</div>
</div>
</div>
<div class="product-wrapper-bundle">
  <div class="product-wrapper mb-30">
    <div class="product-img">
      <a href="product-details.html">
        
      </a>
    <div class="product-item-dec">
      <ul>
        <li>2018</li>
        <li>MANUAL</li>
        <li>PETROL</li>
        <li>150 CC</li>
      </ul>
    </div>
    <div class="product-action">
      <a class="action-plus-2" title="Add To Cart" href="#">
        <i class=" ti-shopping-cart"></i>
      </a>
      <a class="action-cart-2" title="Wishlist" href="#">
        <i class=" ti-heart"></i>
      </a>
      <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
        <i class=" ti-zoom-in"></i>
      </a>
    </div>
  <div class="product-content-wrapper">
    <div class="product-title-spreed">
      <h4><a href="product-details.html">Matrio

      <span>4600 RPM</span>
    </div>
    <div class="product-price">
```

```

        <span>$2549</span>
    </div>
</div>
</div>
</div>
<div class="product-wrapper mb-30">
    <div class="product-img">
        <a href="product-details.html">
            
        </a>
        <div class="product-item-dec">
            <ul>
                <li>2018</li>
                <li>MANUAL</li>
                <li>PETROL</li>
                <li>250 CC</li>
            </ul>
        </div>
        <div class="product-action">
            <a class="action-plus-2" title="Add To Cart" href="#">
                <i class=" ti-shopping-cart"></i>
            </a>
            <a class="action-cart-2" title="Wishlist" href="#">
                <i class=" ti-heart"></i>
            </a>
            <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
                <i class=" ti-zoom-in"></i>
            </a>
        </div>
        <div class="product-content-wrapper">
            <div class="product-title-spreed">
                <h4><a href="product-details.html">Rigoniss Z
1000</a></h4>
                <span>8000 RPM</span>
            </div>
            <div class="product-price">
                <span>$2549</span>
            </div>
        </div>
    </div>
</div>
</div>
<div class="product-wrapper-bundle">
    <div class="product-wrapper mb-30">

```

```

<div class="product-img">
  <a href="product-details.html">
    
  </a>
  <div class="product-item-dec">
    <ul>
      <li>2018</li>
      <li>MANUAL</li>
      <li>PETROL</li>
      <li>250 CC</li>
    </ul>
  </div>
  <div class="product-action">
    <a class="action-plus-2" title="Add To Cart" href="#">
      <i class=" ti-shopping-cart"></i>
    </a>
    <a class="action-cart-2" title="Wishlist" href="#">
      <i class=" ti-heart"></i>
    </a>
    <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
      <i class=" ti-zoom-in"></i>
    </a>
  </div>
  <div class="product-content-wrapper">
    <div class="product-title-spreed">
      <h4><a href="product-details.html">Klager GSX 250
R</a></h4>
      <span>5500 RPM</span>
    </div>
    <div class="product-price">
      <span>$2549</span>
    </div>
  </div>
</div>
<div class="product-wrapper mb-30">
  <div class="product-img">
    <a href="product-details.html">
      
    </a>
    <div class="product-item-dec">
      <ul>
        <li>2018</li>
        <li>MANUAL</li>

```



```

        <li>PETROL</li>
        <li>350 CC</li>
    </ul>
</div>
<div class="product-action">
    <a class="action-plus-2" title="Add To Cart" href="#">
        <i class=" ti-shopping-cart"></i>
    </a>
    <a class="action-cart-2" title="Wishlist" href="#">
        <i class=" ti-heart"></i>
    </a>
    <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
        <i class=" ti-zoom-in"></i>
    </a>
</div>
<div class="product-content-wrapper">
    <div class="product-title-spreed">
        <h4><a href="product-details.html">Maxclon ZPE
54</a></h4>

        <span>3300 RPM</span>
    </div>
    <div class="product-price">
        <span>$2549</span>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="latest-product-area pt-205 pb-145 bg-img" style="background-
image: url(assets/img/banner/banner-4.jpg)">
    <div class="container-fluid">
        <div class="latest-product-slider owl-carousel">
            <div class="single-latest-product slider-animated-2">
                <div class="row">
                    <div class="col-lg-7 col-md-12 col-12">
                        <div class="latest-product-img">
                            
                        </div>

```

```

</div>
<div class="col-lg-5 col-md-12 col-12">
  <div class="latest-product-content">
    <h2 class="animated">LATEST OFFER <br>FOR POPULAR
BIKES</h2>
    <p class="animated"><span>OSWAN</span> the most latgest
bike store in the wold can serve you latest      qulity of motorcycle also you can
sell here your motorcycle it quo minus iduod maxie placeat facere possimus, omnis
voluptas assumenda est, omnis dolor llendus. Temporibus autem quibusdam </p>
    <div class="latest-price">
      <h3 class="animated">NOW AT
<span>$1250</span></h3>
      <span class="animated">35% DISCOUNT</span>
    </div>
    <div class="lertext-btn">
      <a class="animated" href="#">SELECT A BIKE</a>
    </div>
  </div>
</div>
</div>
<div class="single-latest-product slider-animated-2">
  <div class="row">
    <div class="col-lg-7 col-col-12 col-12">
      <div class="latest-product-img">
        
      </div>
    </div>
    <div class="col-lg-5 col-col-12 col-12">
      <div class="latest-product-content">
        <h2 class="animated">LATEST OFFER <br>FOR POPULAR
BIKES</h2>
        <p class="animated"><span>OSWAN</span> the most latgest
bike store in the wold can serve you latest      qulity of motorcycle also you can
sell here your motorcycle it quo minus iduod maxie placeat facere possimus, omnis
voluptas assumenda est, omnis dolor llendus. Temporibus autem quibusdam </p>
        <div class="latest-price">
          <h3 class="animated">NOW AT
<span>$1250</span></h3>
          <span class="animated">35% DISCOUNT</span>
        </div>
        <div class="lertext-btn">
          <a class="animated" href="#">SELECT A BIKE</a>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="accessories-area pt-152 pb-130">
    <div class="container-fluid">
        <div class="section-title section-fluid text-center mb-60">
            <h2>ACCESSORIES</h2>
            <p><span>OSWAN</span> the most latgest bike store in the wold can
serve you latest          qulity of motorcycle also you can sell here your
motorcycle</p>
        </div>
        <div class="accessories-wrapper">
            <div class="product-accessories-active owl-carousel">
                <div class="product-wrapper">
                    <div class="product-img">
                        <a href="#">
                            
                        </a>
                    <div class="product-action">
                        <a class="action-plus-2" title="Add To Cart" href="#">
                            <i class=" ti-shopping-cart"></i>
                        </a>
                        <a class="action-cart-2" title="Wishlist" href="#">
                            <i class=" ti-heart"></i>
                        </a>
                        <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
                            <i class=" ti-zoom-in"></i>
                        </a>
                    </div>
                <div class="product-content-wrapper-2">
                    <div class="product-title-price-2 text-center">
                        <span>Price: $120</span>
                        <h4><a href="product-details.html">Aerion Carbon
Helmet</a></h4>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="product-wrapper">
        <div class="product-img">

```

```

<a href="product-details.html">
  
</a>
<div class="product-action">
  <a class="action-plus-2" title="Add To Cart" href="#">
    <i class=" ti-shopping-cart"></i>
  </a>
  <a class="action-cart-2" title="Wishlist" href="#">
    <i class=" ti-heart"></i>
  </a>
  <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
    <i class=" ti-zoom-in"></i>
  </a>
</div>
<div class="product-content-wrapper-2">
  <div class="product-title-price-2 text-center">
    <span>Price: $180</span>
    <h4><a href="product-details.html">Reckles
Jacket</a></h4>
  </div>
</div>
</div>
<div class="product-wrapper">
  <div class="product-img">
    <a href="product-details.html">
      
    </a>
  <div class="product-action">
    <a class="action-plus-2" title="Add To Cart" href="#">
      <i class=" ti-shopping-cart"></i>
    </a>
    <a class="action-cart-2" title="Wishlist" href="#">
      <i class=" ti-heart"></i>
    </a>
    <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
      <i class=" ti-zoom-in"></i>
    </a>
  </div>
  <div class="product-content-wrapper-2">
    <div class="product-title-price-2 text-center">
      <span>Price: $25</span>

```

```

        <h4><a href="product-details.html">Softy Original
Glove</a></h4>
        </div>
    </div>
</div>
<div class="product-wrapper">
    <div class="product-img">
        <a href="product-details.html">
            
        </a>
        <div class="product-action">
            <a class="action-plus-2" title="Add To Cart" href="#">
                <i class=" ti-shopping-cart"></i>
            </a>
            <a class="action-cart-2" title="Wishlist" href="#">
                <i class=" ti-heart"></i>
            </a>
            <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
                <i class=" ti-zoom-in"></i>
            </a>
        </div>
        <div class="product-content-wrapper-2">
            <div class="product-title-price-2 text-center">
                <span>Price: $140</span>
                <h4><a href="product-details.html">Flicky Traco
Boot</a></h4>
            </div>
        </div>
    </div>
</div>
<div class="product-wrapper">
    <div class="product-img">
        <a href="product-details.html">
            
        </a>
        <div class="product-action">
            <a class="action-plus-2" title="Add To Cart" href="#">
                <i class=" ti-shopping-cart"></i>
            </a>
            <a class="action-cart-2" title="Wishlist" href="#">
                <i class=" ti-heart"></i>
            </a>
        </div>
    </div>

```

```

        <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
        <i class=" ti-zoom-in"></i>
    </a>
</div>
<div class="product-content-wrapper-2">
    <div class="product-title-price-2 text-center">
        <span>Price: $150</span>
        <h4><a href="product-details.html">Flicky Traco
Boot</a></h4>
    </div>
</div>
</div>
</div>
<div class="product-wrapper">
    <div class="product-img">
        <a href="product-details.html">
            
        </a>
    <div class="product-action">
        <a class="action-plus-2" title="Add To Cart" href="#">
            <i class=" ti-shopping-cart"></i>
        </a>
        <a class="action-cart-2" title="Wishlist" href="#">
            <i class=" ti-heart"></i>
        </a>
        <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
            <i class=" ti-zoom-in"></i>
        </a>
    </div>
<div class="product-content-wrapper-2">
    <div class="product-title-price-2 text-center">
        <span>Price: $150</span>
        <h4><a href="product-details.html">Aerion Carbon
Helmet</a></h4>
    </div>
</div>
</div>
</div>
<div class="product-wrapper">
    <div class="product-img">
        <a href="product-details.html">
            
        </a>

```

```

<div class="product-action">
  <a class="action-plus-2" title="Add To Cart" href="#">
    <i class=" ti-shopping-cart"></i>
  </a>
  <a class="action-cart-2" title="Wishlist" href="#">
    <i class=" ti-heart"></i>
  </a>
  <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
    <i class=" ti-zoom-in"></i>
  </a>
</div>
<div class="product-content-wrapper-2">
  <div class="product-title-price-2 text-center">
    <span>Price: $160</span>
    <h4><a href="product-details.html">Reckles
Jacket</a></h4>
  </div>
</div>
</div>
<div class="product-wrapper">
  <div class="product-img">
    <a href="product-details.html">
      
    </a>
  <div class="product-action">
    <a class="action-plus-2" title="Add To Cart" href="#">
      <i class=" ti-shopping-cart"></i>
    </a>
    <a class="action-cart-2" title="Wishlist" href="#">
      <i class=" ti-heart"></i>
    </a>
    <a class="action-reload" title="Quick View" data-
toggle="modal" data-target="#exampleModal" href="#">
      <i class=" ti-zoom-in"></i>
    </a>
  </div>
  <div class="product-content-wrapper-2">
    <div class="product-title-price-2 text-center">
      <span>Price: $120</span>
      <h4><a href="product-details.html">Aerion Carbon
Helmet</a></h4>
    </div>
  </div>
</div>

```

```

        </div>
    </div>
</div>
</div>
</div>
<div class="testimonial-area">
    <div class="container">
        <div class="section-title-2 section-title-position">
            <h2>OUR CLIENTS REVIEW</h2>
        </div>
        <div class="testimonial-active owl-carousel">
            <div class="single-testimonial">
                <div class="row">
                    <div class="col-lg-5">
                        <div class="testimonial-img pl-75">
                            
                        </div>
                    </div>
                    <div class="col-lg-7">
                        <div class="testimonial-content">
                            <div class="testimonial-dec">
                                <p><span>OSWAN</span> the most latgest bike store in
the wold can serve you latest  quality of motorcycle also you can sell here your
motorcycle it quo minus iduod maxie placeat facere possimus, omnis voluptas
assumenda est, omnis dolor llendus. Temporibus autem quibusdam quoten</p>
                            </div>
                            <div class="name-designation">
                                <h4>Rayed Ayash Hisham</h4>
                                <span>COO, ASEKHA</span>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="single-testimonial">
    <div class="row">
        <div class="col-lg-5 col-md-12 col-12">
            <div class="testimonial-img pl-75">
                
            </div>
        </div>
        <div class="col-lg-7 col-md-12 col-12">
            <div class="testimonial-content">
                <div class="testimonial-dec">

```



<p><span>OSWAN</span> Lorem ipsum dolor sit amet, consectetur adipisicing , sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex commodo consequat. Duis dolor in reprehenderit.</p>

</div>

<div class="name-designation">

<h4>James Momen Nirob</h4>

<span>CEO, ASEKHA</span>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="blog-area pt-150 pb-110">

<div class="container">

<div class="section-title text-center mb-60">

<h2>BLOG POST</h2>

<p><span>OSWAN</span> the most latgest bike store in the wold can serve you latest qulity of motorcycle also you can sell here your motorcycle</p>

</div>

<div class="row">

<div class="col-lg-4 col-md-6">

<div class="blog-hm-wrapper mb-40">

<div class="blog-img">

<a href="blog-details.html"></a>

<div class="blog-date">

<h4>24 February, 2018</h4>

</div>

<div class="blog-hm-social">

<ul>

<li><a href="#"><i class="fa fa-facebook"></i></a></li>

<li><a href="#"><i class="fa fa-twitter"></i></a></li>

<li><a href="#"><i class="fa fa-google-plus"></i></a></li>

</ul>

</div>

</div>

<div class="blog-hm-content">

<h3><a href="blog-details.html">Sports Motorbike for play in desert </a></h3>

<p><span>OSWAN</span> the most latgest bike store in the  
wold can serve you latest quility of motorcycle soucan sell here your motorcycle it  
quo </p>

</div>

</div>

</div>

<div class="col-lg-4 col-md-6">

<div class="blog-hm-wrapper mb-40">

<div class="blog-img">

<a href="blog-details.html"></a>

<div class="blog-date">

<h4>22 February, 2018</h4>

</div>

<div class="blog-hm-social">

<ul>

<li><a href="#"><i class="fa fa-facebook"></i></a></li>

<li><a href="#"><i class="fa fa-twitter"></i></a></li>

<li><a href="#"><i class="fa fa-google-plus"></i></a></li>

</ul>

</div>

</div>

<div class="blog-hm-content">

<h3><a href="blog-details.html">Motorbike Racing at  
October</a></h3>

<p><span>OSWAN</span> the most latgest bike store in the  
wold can serve you latest quility of motorcycle soucan sell here your motorcycle it  
quo </p>

</div>

</div>

</div>

<div class="col-lg-4 col-md-6">

<div class="blog-hm-wrapper mb-40">

<div class="blog-img">

<a href="blog-details.html"></a>

<div class="blog-date">

<h4>20 February, 2018</h4>

</div>

<div class="blog-hm-social">

<ul>

<li><a href="#"><i class="fa fa-facebook"></i></a></li>

<li><a href="#"><i class="fa fa-twitter"></i></a></li>

<li><a href="#"><i class="fa fa-google-plus"></i></a></li>

</ul>

```

        </div>
    </div>
    <div class="blog-hm-content">
        <h3><a href="blog-details.html">Latest Motorbike Release this
Year </a></h3>
        <p><span>OSWAN</span> the most latgest bike store in the
wold can serve you latest qulity of motorcycle soucan sell here your motorcycle it
quo </p>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="newsletter-area">
    <div class="container">
        <div class="newsletter-wrapper-all theme-bg-2">
            <div class="row">
                <div class="col-lg-5 col-12 col-md-12">
                    <div class="newsletter-img bg-img" style="background-image:
url(assets/img/banner/newsletter-bg.png)">
                        
                    </div>
                </div>
                <div class="col-lg-7 col-12 col-md-12">
                    <div class="newsletter-wrapper text-center">
                        <div class="newsletter-title">
                            <h3>Subscribe our newsletter</h3>
                        </div>
                        <div id="mc_embed_signup" class="subscribe-form">
                            <form action="#" method="post" id="#" name="mc-
embedded-subscribe-form" class="validate" target="_blank" novalidate>
                                <div id="mc_embed_signup_scroll" class="mc-form">
                                    <input type="email" value="" name="EMAIL"
class="email" placeholder="Enter your email here..." required>
                                    <!-- real people should not fill this in and expect good
things - do not remove this or risk form bot signups-->
                                    <div class="mc-news" aria-hidden="true"><input
type="text" name="b_6bbb9b6f5827bd842d9640c82_05d85f18ef" tabindex="-1"
value=""></div>
                                    <div class="clear"><input type="submit"
value="Subscribe" name="subscribe" id="mc-embedded-subscribe"
class="button"></div>
                                </div>
                            </form>

```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="footer-top pt-210 pb-98 theme-bg">
    <div class="container">
        <div class="row">
            <div class="col-lg-3 col-md-6 col-12">
                <div class="footer-widget mb-30">
                    <div class="footer-logo">
                        <a href="index.html">
                            
                        </a>
                    </div>
                    <div class="footer-about">
                        <p><span>OSWAN</span> the most latgest bike store in the
wold can serve you latest ulity of motorcycle soucan sell here your motorcycle it quo
</p>
                        <div class="footer-support">
                            <h5>FOR SUPPORT</h5>
                            <span> 01245 658 698 (Toll Free)</span>
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-lg-3 col-md-6 col-12">
                <div class="footer-widget mb-30 pl-60">
                    <div class="footer-widget-title">
                        <h3>QUICK LINK</h3>
                    </div>
                    <div class="quick-links">
                        <ul>
                            <li><a href="about-us.html">About us</a></li>
                            <li><a href="#">Service</a></li>
                            <li><a href="#">Inventory</a></li>
                            <li><a href="shop.html">Shop</a></li>
                            <li><a href="blog-sidebar.html">Blog</a></li>
                            <li><a href="#">Conditions</a></li>
                            <li><a href="contact.html">Contact</a></li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

    </div>
  </div>
  <div class="col-lg-3 col-md-6 col-12">
    <div class="footer-widget mb-30">
      <div class="footer-widget-title">
        <h3>LATEST TWEET</h3>
      </div>
      <div class="food-widget-content pr-30">
        <div class="single-tweet">
          <p>
            the most latgest bike store in the wold can serve you
            10 min ago
          </p>
        </div>
        <div class="single-tweet">
          <p>
            the most latgest bike store in the wold can serve you
            10 min ago
          </p>
        </div>
        <div class="single-tweet">
          <p>
            the most latgest bike store in the wold can serve you
            10 min ago
          </p>
        </div>
      </div>
    </div>
  </div>
  <div class="col-lg-3 col-md-6 col-12">
    <div class="footer-widget mb-30">
      <div class="footer-widget-title">
        <h3>CONTACT INFO</h3>
      </div>
      <div class="food-info-wrapper">
        <div class="food-address">
          <div class="food-info-title">
            <span>Address</span>
          </div>
          <div class="food-info-content">
            <p>276 Jhilli Nogor, 4th folor, Momen Tower, Main
Town, New Yourk</p>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```
<div class="food-info-title">  
    <span>Phone</span>  
</div>  
<div class="food-info-content">  
    <p>+090 12568 369 987</p>  
    <p>+090 12568 369 987</p>  
</div>  
</div>  
<div class="food-address">  
    <div class="food-info-title">  
        <span>Web</span>  
    </div>  
    <div class="food-info-content">  
        <a href="#">info@oswanmega.com</a>  
        <a href="#">www.oswanmega.com</a>  
    </div>  
</div>  
</div>  
</div>  
</div>  
</div>  
</div>  
<div class="footer-bottom ptb-35 black-bg">  
    <div class="container">  
        <div class="row">  
            <div class="col-md-8 col-12">  
                <div class="copyright">  
                    <p>©Copyright, 2018 All Rights Reserved by <a  
href="https://freethemescloud.com/">Free themes Cloud</a></p>  
                </div>  
            </div>  
            <div class="col-md-4 col-12">  
                <div class="footer-payment-method">  
                    <a href="#"></a>  
                </div>  
            </div>  
        </div>  
    </div>  
</div>  
</div>  
</div>  
</div>  
</div>  
</div>
```

```

        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
        <span class="icofont icofont-close" aria-hidden="true"></span>
</button>
<div class="modal-dialog" role="document">
    <div class="modal-content">
        <div class="modal-body">
            <div class="qwick-view-left">
                <div class="quick-view-learg-img">
                    <div class="quick-view-tab-content tab-content">
                        <div class="tab-pane active show fade" id="modal1"
role="tabpanel">
                            
                        </div>
                        <div class="tab-pane fade" id="modal2" role="tabpanel">
                            
                        </div>
                        <div class="tab-pane fade" id="modal3" role="tabpanel">
                            
                        </div>
                    </div>
                </div>
                <div class="quick-view-list nav" role="tablist">
                    <a class="active" href="#modal1" data-toggle="tab" role="tab">
                        
                    </a>
                    <a href="#modal2" data-toggle="tab" role="tab">
                        
                    </a>
                    <a href="#modal3" data-toggle="tab" role="tab">
                        
                    </a>
                </div>
            </div>
            <div class="qwick-view-right">
                <div class="qwick-view-content">
                    <h3>Aerion Carbon Helmet</h3>
                    <div class="price">
                        <span class="new">$90.00</span>
                        <span class="old">$120.00 </span>
                    </div>
                    <div class="rating-number">
                        <div class="quick-view-rating">
                            <i class="fa fa-star reting-color"></i>
                            <i class="fa fa-star reting-color"></i>

```

```

        <i class="fa fa-star reting-color"></i>
        <i class="fa fa-star reting-color"></i>
        <i class="fa fa-star reting-color"></i>
    </div>
</div>
<p>Lorem ipsum dolor sit amet, consectetur adip elit, sed do
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad mi , quis nostrud
veniam exercitation .</p>
<div class="quick-view-select">
    <div class="select-option-part">
        <label>Size*</label>
        <select class="select">
            <option value="">- Please Select -</option>
            <option value="">900</option>
            <option value="">700</option>
        </select>
    </div>
    <div class="select-option-part">
        <label>Color*</label>
        <select class="select">
            <option value="">- Please Select -</option>
            <option value="">orange</option>
            <option value="">pink</option>
            <option value="">yellow</option>
        </select>
    </div>
</div>
<div class="quickview-plus-minus">
    <div class="cart-plus-minus">
        <input type="text" value="02" name="qtybutton"
class="cart-plus-minus-box">
    </div>
    <div class="quickview-btn-cart">
        <a class="btn-style" href="#">add to cart</a>
    </div>
    <div class="quickview-btn-wishlist">
        <a class="btn-hover" href="#"><i class="icofont icofont-
heart-alt"></i></a>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>

```



```
</div>
</div>
```

```
<!-- all js here -->
<script src="assets/js/vendor/jquery-1.12.0.min.js"></script>
<script src="assets/js/popper.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script src="assets/js/isotope.pkgd.min.js"></script>
<script src="assets/js/imagesloaded.pkgd.min.js"></script>
<script src="assets/js/jquery.counterup.min.js"></script>
<script src="assets/js/waypoints.min.js"></script>

<script src="assets/js/owl.carousel.min.js"></script>
<script src="assets/js/plugins.js"></script>
<script src="assets/js/main.js"></script>
</body>
</html>
```

## User folder

### Create.cshtml

```
@model Application.DTO.UserDTO
```

```
@{
    ViewData["Title"] = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

```
<h2>Create</h2>
```

```
<h4>UserDTO</h4>
```

```
<hr />
```

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

```
    <div class="col-md-4">
```

```

<form asp-action="Create">
  <div asp-validation-summary="ModelOnly" class="text-danger"></div>
  <div class="form-group">
    <label asp-for="Userid" class="control-label"></label>
    <input asp-for="Userid" class="form-control" />
    <span asp-validation-for="Userid" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="Username" class="control-label"></label>
    <input asp-for="Username" class="form-control" />
    <span asp-validation-for="Username" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="Email" class="control-label"></label>
    <input asp-for="Email" class="form-control" />
    <span asp-validation-for="Email" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="Password" class="control-label"></label>
    <input asp-for="Password" class="form-control" />
    <span asp-validation-for="Password" class="text-danger"></span>
  </div>
  <div class="form-group">
    <input type="submit" value="Create" class="btn btn-default" />
  </div>
</form>
</div>
</div>

<div>
  <a asp-action="Index">Back to List</a>
</div>

```

## Delete.cshtml

```
@model Application.DTO.UserDTO
```

```

@{
    ViewData["Title"] = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```
<h2>Delete</h2>
```

```

<h3>Are you sure you want to delete this?</h3>
<div>
  <h4>UserDTO</h4>
  <hr />
  <dl class="dl-horizontal">
    <dt>
      @Html.DisplayNameFor(model => model.Userid)
    </dt>
    <dd>
      @Html.DisplayFor(model => model.Userid)
    </dd>
    <dt>
      @Html.DisplayNameFor(model => model.Username)
    </dt>
    <dd>
      @Html.DisplayFor(model => model.Username)
    </dd>
    <dt>
      @Html.DisplayNameFor(model => model.Email)
    </dt>
    <dd>
      @Html.DisplayFor(model => model.Email)
    </dd>
    <dt>
      @Html.DisplayNameFor(model => model.Password)
    </dt>
    <dd>
      @Html.DisplayFor(model => model.Password)
    </dd>
  </dl>

  <form asp-action="Delete">
    <input type="submit" value="Delete" class="btn btn-default" /> |
    <a asp-action="Index">Back to List</a>
  </form>
</div>

```

## Details.cshtml

```

@model Application.DTO.UserDTO

@{
    ViewData["Title"] = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

<h2>Details</h2>

<div>

<h4>UserDTO</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.Userid)

</dt>

<dd>

@Html.DisplayFor(model => model.Userid)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Username)

</dt>

<dd>

@Html.DisplayFor(model => model.Username)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Email)

</dt>

<dd>

@Html.DisplayFor(model => model.Email)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Password)

</dt>

<dd>

@Html.DisplayFor(model => model.Password)

</dd>

</dl>

</div>

<div>

@Html.ActionLink("Edit", "Edit", new { /\* id = Model.PrimaryKey \*/ }) |

<a asp-action="Index">Back to List</a>

</div>

## Edit.cshtml

@model Application.DTO.UserDTO

@{

ViewData["Title"] = "Edit";

Layout = "~/Views/Shared/\_Layout.cshtml";

```
}
```

```
<h2>Edit</h2>
```

```
<h4>UserDTO</h4>
```

```
<hr />
```

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

```
  <div class="col-md-4">
```

```
    <form asp-action="Edit">
```

```
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
      <div class="form-group">
```

```
        <label asp-for="Userid" class="control-label"></label>
```

```
        <input asp-for="Userid" class="form-control" />
```

```
        <span asp-validation-for="Userid" class="text-danger"></span>
```

```
      </div>
```

```
      <div class="form-group">
```

```
        <label asp-for="Username" class="control-label"></label>
```

```
        <input asp-for="Username" class="form-control" />
```

```
        <span asp-validation-for="Username" class="text-danger"></span>
```

```
      </div>
```

```
      <div class="form-group">
```

```
        <label asp-for="Email" class="control-label"></label>
```

```
        <input asp-for="Email" class="form-control" />
```

```
        <span asp-validation-for="Email" class="text-danger"></span>
```

```
      </div>
```

```
      <div class="form-group">
```

```
        <label asp-for="Password" class="control-label"></label>
```

```
        <input asp-for="Password" class="form-control" />
```

```
        <span asp-validation-for="Password" class="text-danger"></span>
```

```
      </div>
```

```
      <div class="form-group">
```

```
        <input type="submit" value="Save" class="btn btn-default" />
```

```
      </div>
```

```
    </form>
```

```
  </div>
```

```
</div>
```

```
<div>
```

```
  <a asp-action="Index">Back to List</a>
```

```
</div>
```

## Index.cshtml

@model Application.DTO.ProductDTO

```
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

<h2>Index</h2>

<p>

<a asp-action="Create">Create New</a>

</p>

<div>

<h4>ProductDTO</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.Id)

</dt>

<dd>

@Html.DisplayFor(model => model.Id)

</dd>

<dt>

@Html.DisplayNameFor(model => model.ProductName)

</dt>

<dd>

@Html.DisplayFor(model => model.ProductName)

</dd>

<dt>

@Html.DisplayNameFor(model => model.SupplierId)

</dt>

<dd>

@Html.DisplayFor(model => model.SupplierId)

</dd>

<dt>

@Html.DisplayNameFor(model => model.UnitPrice)

</dt>

<dd>

@Html.DisplayFor(model => model.UnitPrice)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Package)

```

        </dt>
        <dd>
            @Html.DisplayFor(model => model.Package)
        </dd>
        <dt>
            @Html.DisplayNameFor(model => model.IsDiscontinued)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.IsDiscontinued)
        </dd>
    </dl>
</div>
<div>
    @Html.ActionLink("Edit", "Edit", new { /* id = Model.PrimaryKey */ }) |
    <a asp-action="Index">Back to List</a>
</div>

```

## Startup.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Application.Commands;
using Application.Commands.User;
using Application.Interface;
using EfCommands;
using EfCommands.User;
using EfCommands.UserEF;
using EfDataAccess;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;

namespace Web
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {

```

```
    Configuration = configuration;
}
```

```
public IConfiguration Configuration { get; }
```

// This method gets called by the runtime. Use this method to add services to the container.

```
public void ConfigureServices(IServiceCollection services)
{
    services.Configure<CookiePolicyOptions>(options =>
    {
        // This lambda determines whether user consent for non-essential cookies is
        // needed for a given request.
        options.CheckConsentNeeded = context => true;
        options.MinimumSameSitePolicy = SameSiteMode.None;
    });
}
```

```
services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
services.AddDbContext<ProjContext>();
services.AddTransient<IAddProductCommand, EfAddProductCommand>();
/// services.AddTransient<IGetProductsCommand,
EfGetProductsCommand>();
services.AddTransient<IGetProductCommand, EfGetProductCommand>();
services.AddTransient<IEditProductCommand, EfEditProductCommand>();
services.AddTransient<IDeleteProductCommand,
EfDeleteProductCommand>();
```

```
services.AddTransient<IAddUserCommand, EfAddUserCommand>();
/// services.AddTransient<IGetProductsCommand,
EfGetProductsCommand>();
services.AddTransient<IGetUserCommand,
EfCommands.UserEF.EfGetUserCommand>();
services.AddTransient<IEditUserCommand, EfEditUserCommand>();
services.AddTransient<IDeleteUserCommand, EfDeleteUserCommand>();
}
```

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
}
```



```
}
else
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseCookiePolicy();

app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller=Home}/{action=Index}/{id?}");
});
}
}
```