**Документация**

**1. Архитектура**

Приложението е структурирано по MVC (Model-View-Controller) модел, като е разширено със сервизен слой, който капсулира бизнес логиката, и база данни, използваща Entity Framework Core.

**2. Преглед на слоевете**

- **Контролери (Controllers)**: Обработват заявките и отговорите на потребителите.   
- **База данни (Data)**: Използва Entity Framework Core за ORM и SQL Server като доставчик на базата данни.  
- **Модели (Models)**: Дефинират структурата на данните и връзките между тях.   
- **Услуги (Services)**: Съдържат бизнес логиката и взаимодействат с базата данни.   
**- Изглед модели (ViewModel):** Осигурява преноса на данни между контролерите и изгледите, като улеснява представянето им към потребителя.  
- **Изгледи (Views)**: Осигуряват потребителския интерфейс

**3. Описание на основните контролери**

**- AboutUsController:** Показва страницата „About us“.

**- AccountController:** Управлява автентикацията на потребителите (вход, регистрация, управление на пароли).

**- AdminController:** Осигурява административни функции като управление на потребители, продукти и поръчки.

**- AdminDashboardController:** Контролер за администраторския панел, достъпен само за администратори.  
**- ApplicationUserManager:** Клас за управление на потребителски роли.

- **CartController:** Управлява добавяне, премахване и показване на продукти в количката на потребителя.

- **ChatController:** Управлява изпращането на съобщения в чат по тикети между потребители и администратори.

- **CouponController:** Създава, показва и управлява промоционални купони.  
**- FavoritesController:** Управлява любими продукти на потребителя – добавяне, премахване и показване.  
- **HomeController:** Зарежда началната страница, показва форума и обработва грешки.

- **HttpStatusCodeResult:** представлява резултат от действие, който връща специфичен HTTP статус код и съобщение.

**- OrderController:** Управлява поръчките на потребителите, историята на поръчките и тяхното състояние.  
- **ProductDetailsController:** Обработва заявките за показване на подробности за продукт, подаване на отзиви и премахване на отзиви, като включва свързани продукти и рецензии.

**- ProductController:** Отговаря за показване на продукти, детайли и търсене.

**- PurHistoryController:** Обработва заявките за показване на история на поръчките на потребителя, като използва услугата за поръчки и показва съответното съдържание.

-**PurchaseHistoryAdminController:** Управлява поръчките на потребителите от административния панел, като позволява преглед, редактиране, добавяне и премахване на продукти от поръчките.

- **StoreController:** Управлява продуктовия каталог на магазина, като предлага функции за търсене, филтриране по категория и сортиране по цена, както и пагинация за показване на продуктите.

- **TicketController:** Управлява билетите за клиентска поддръжка, като позволява на потребителите да създават и проследяват заявки.

**4. Роли на потребителите и техните права**

**- Потребител:** може да разглежда продукти, добавя в количка, извършва поръчки, пише отзиви, създава тикети.

**- Администратор:** имапълен достъп до всички функционалности, включително преглед и отговор на тикети, добавяне на продукти, управление на купони.

**5. Модели**

**- CartItem:** Представлява елемент в количката, включващ продукт и количеството му.

**- Coupon:** Представлява промоционален код с отстъпка, който има ограничение за използванията и валидност.

**- ErrorViewModel:** Съдържа информация за идентификатора на заявка и дали той трябва да бъде показан в изгледа на грешката.  
- **FavoriteProduct:** Представлява модел за съхранение на любими продукти на потребител, свързани с конкретен продукт и потребител.  
- **Location:** Представлява модел за съхранение на информация за местоположение, включително име, географски координати и описание.

- **Message:** Представлява съобщение, свързано с тикет, изпратено от потребител или администратор, съдържащо съдържание и дата на изпращане.

**- Product:** Представлява продукт със свойства като име, марка, категория, цена, описание, количество и изображения, като може да съдържа отзиви.

- **ProductDto:** Представлява данни за продукт, включително име, марка, категория, цена, описание, количество и файл за изображение, които се използват при създаване или обновяване на продукт.

- **Review:** Представлява рецензия за продукт с автор, съдържание, оценка (между 1 и 5 звезди) и дата на създаване.

- **Ticket:** Модел за управление на тикети, съдържащ информация за потребителя, администратор, заглавие, описание, статус (Open, InProgress, Resolved, Closed) и съобщения.

- **Users:** Разширява класата IdentityUser, добавяйки поле за пълното име на потребителя.

**6. Услуги**

- **OrderService** – връща поръчките на потребителя.

- **ChatService** – връща и изпраща съобщения по тикет.

- **CouponService** – проверява и създава купони.

- **FavorteProductService** – добавя и премахва любими продукти.

- **TicketService** – създава, затваря и асоциира тикети.

**7. View Модели**

**- CartItemViewModel** – артикул в количката.

**- CartViewModel** – количка с потребителски данни.

**- ChangePasswordViewModel** – форма за смяна на парола.

**- EditUserViewModel** – форма за редакция на потребител.

**- HomeViewModel** – информация на началната страница.

**- LoginViewModel** – форма за вход.

**- Order** -модел, съдържащ данни за поръчка, клиент и свързани продукти.

**- OrderConfirmationViewModel** – потвърждение на поръчка.

**- ProductDetailViewModel** – детайли за продукт с отзиви.

**- ProductDetailsViewModel** – продукт и свързани продукти.

**- RegisterViewModel** – регистрационна форма.

**- UserProfileViewModel** – профил и промяна на парола.

**- UserViewModel** – основна информация за потребител.

**- VerifyEmailViewModel** – форма за потвърждение на имейл.