

# Discounts.ItAcademy.Ge

## 1. სისტემის როლები და ავტორიზაცია

აპლიკაციას უნდა ჰქონდეს როლებზე დაფუძნებული წვდომა (**Role-based Authorization**):

- **Administrator:** სისტემის გლობალური მმართველი.
- **Merchant (Seller):** კომპანია/ორგანიზაცია, რომელიც განათავსებს ფასდაკლებებს.
- **Customer:** ფიზიკური პირი, რომელიც ყიდულობს კუპონებს.

## 2. როლების ფუნქციონალი (MVC & API)

### 2.1 მერჩანტის (Merchant) მოდული

ეს არის ორგანიზაციის პანელი, სადაც ის მართავს საკუთარ შეთავაზებებს:

- **Dashboard:** ხედავს თავისი აქტიური და ვადაგასული ფასდაკლებების სტატისტიკას.
- **Offer CRUD:** შეუძლია ახალი ფასდაკლების დამატება (სათაური, აღწერა, კატეგორია, სურათი, ორიგინალი ფასი, ფასდაკლებული ფასი, კუპონების რაოდენობა, ვადები).
  - **შენიშვნა:** დამატებული ფასდაკლება სტატუსით არის "Pending", სანამ ადმინი არ დაამტკიცებს.
- **Edit Function:** შეუძლია შეცვალოს დეტალები მხოლოდ X პერიოდის განმავლობაში (ადმინის მიერ დაწესებული ლიმიტი).
- **Sales History:** ხედავს ვინ იყიდა მისი კუპონები (მომხმარებლის სახელი, თარიღი, კოდი).

### 2.2 ადმინისტრატორის (Admin) მოდული

სისტემის სრული კონტროლი:

- **User Management:** მერჩანტებისა და მომხმარებლების სრული CRUD (შექმნა, დაბლოკვა, რედაქტირება).
- **Moderation:** ხედავს "Pending" სტატუსის მქონე ფასდაკლებებს და აქვს ორი ღილაკი: **Approve** (გამოაჩენს საიტზე) და **Reject** (უარყოფს მიზეზის მითითებით).
- **Global Settings:** გვერდი, სადაც ადმინი განსაზღვრავს სისტემურ პარამეტრებს:
  - ჯავშნის ხანგრძლივობა (მაგ: 30 წუთი).
  - რედაქტირების ვადა მერჩანტისთვის (მაგ: 24 საათი).
- **Category Management:** ფასდაკლების კატეგორიების მართვა (რესტორნები, ტურიზმი და ა.შ.).

### 2.3 მომხმარებლის (Customer) მოდული

- **Browse & Search:** ათვალიერებს დამტკიცებულ ფასდაკლებებს კატეგორიების და ფასის მიხედვით.
  - **Booking & Purchase:**
    - **დაჯავშნა:** აკეთებს დროებით რეზერვაციას. ამ დროს კუპონების რაოდენობა აკლდება ნაშთს. თუ **X** დროში არ იყიდის, **Worker**-ი შლის ჯავშანს.
    - **ყიდვა:** ყიდულობს კუპონს (სიმულაცია, ბალანსის დაკლება ან უბრალოდ დასტური).
  - **My Coupons:** მომხმარებლის პირადი გვერდი, სადაც ხედავს მის მიერ შეძენილ ყველა კუპონს და მათ სტატუსს (აქტიური/გამოყენებული/ვადაგასული).
- 

### 3. ფასდაკლების დეტალების გვერდი (Public)

თითოეულ ფასდაკლებაზე **Details** ღილაკით გადასვლისას:

- ჩანს სრული ინფო, დარჩენილი რაოდენობა და ვადები.
  - თუ მომხმარებელი ავტორიზებულია როგორც **Customer**, უჩანს "დაჯავშნა" და "ყიდვა".
  - თუ ნაშთი 0-ია, ღილაკები იცვლება ტექსტით "გაყიდულია".
- 

### 4. Background Tasks (Worker Service)

სისტემამ ავტომატურად, ინტერვალურად უნდა შეასრულოს:

1. **Reservation Cleanup:** წაშალოს ის ჯავშნები, რომლებსაც გაუვიდათ ადმინის მიერ დაწესებული დრო.
  2. **Offer Expiration:** მთავარი გვერდიდან აიღოს (სტატუსი შეუცვალოს) ის ფასდაკლებები, რომელთა მოქმედების ვადაც ამოიწურა.
- 

### ტექნიკური მოთხოვნები

1. **SQL Server & EF Core (Code First).**
2. **API & MVC** ერთ Solution-ში (გაყოფილი Presentation Layer).
3. **Authentication:** Identity Framework ან JWT (API-სთვის).
4. **Mapper:** Mapster/AutoMapper (DTO და ViewModel გამიჯვნა).
5. **Validation:** FluentValidation (სავალდებულო ველების ვალიდაციისთვის).
6. **Error Handling:** Global Exception Middleware.
7. **Middlewares**
8. **Swagger**
9. **Api versioning**
10. **Health checks**

11. **Asynchronous methods & CancellationTokens**
12. **Client side & server side validations**
13. **Unit testing – for application layer**
14. **OOP**
15. **SOLID**
16. **Repository Pattern**
17. **.editorconfig**

Optional:

1. Paging
2. CreateBootstrapLogger, Serilog
3. MediatR