

# Events.ItAcademy.Ge

## 1. Events.ItAcademy.Ge – MVC აპლიკაცია

1.2 საიტს გააჩნია რეგისტრაცია/ავტორიზაცია

1.3 რეგისტრაცია/ავტორიზაციის გავლის შემდეგ მომხმარებელს უჩანს „დაამატე ივენთი“ (ასევე სხვა მომხმარებლების მიერ დამატებული ივენთები, რომლებიც ადმინისტრატორმა გამოაჩინა მთვარ გვერდზე) რომელზე დაჭერის შედეგაც გადადის ფორმაზე საიდანაც ქმნის ახალ ივენთს მაგ: „24 სექტემბერი - ჯაზ სალამო მუშტაიდის ბალში“ ივენთის სავალდებულო ველებია: სათაური, აღწერა, ბილეთების რაოდენობა, დაწყების თარიღი, დასრულების დარიღი. (დანარჩენი ველები თავად გადანყვიტეთ მაგ: სურათი და ა.შ.) შექმნილი ივენთი უჩანს ადმინისტრატორს შესაბამის პანელში და ის ნყვეტს გამოაჩნოს თუ არა საჯაროდ.

1.4 იმ შემთხვევაში თუ მომხმარებელი არ ავტორიზდება სისტემაში მას გააჩნია მხოლოდ ხედვის უფლება, ( ათვალერებს არსებულ ივენთებს)

1.5 თითოეულ ივენთზე - „Details“ ლილაკზე დაჭერისას სისტემა უნდა გადავიდეს ახალ გვერდზე სადაც გამოჩნდება დეტალური ინფორმაცია ივენთზე, (ცხრილში უნდა ჩანდეს მხოლოდ რამოდენიმე ველი : დასახელება, თარიღი და ა.შ ხოლო დეტალებზე დაჭერისას სრული ინფო)

1.5.1 დეტალების გვერდზე ასევე უნდა ჩანდეს ლილაკი რომელიც იძლევა საშუალებას ღონისძიებაზე ბილეთის დაჯავშნის(მხოლოდ იმ შემთხვევაში თუ მომხმარებელი ავტორიზებულია). დაჯავშნის მომენტიდან მომხმარებელს ეძლევა X პერიოდი (რომელიც რეგულირდება ადმინისტრატორის პანელიდან ადმინის მიერ) რათა შეიძინოს ბილეთი, X პერიოდის გასვლის შემდეგ თუ მომხმარებელმა არ შეიძინა დაჯავშნილი ბილეთი სისტემამ ავტომატურად უნდა მოხსნას ჯავშანი (worker ის საშუალებით) დაჯავშნის ლილაკზე დაჭერისას სისტემა აყენებს ჯავშნის სტატუსს კონკრეტულ ივენთზე, რაც გამოიხატება იმაში, რომ დროებით 1 -ით ნაკლები ხელმისაწვდომი ბილეთი იქნება სხვა მომხმარებელისათვის.

თუ მომხმარებელმა სისტემის მიერ მოცემულ ჯავშნის პერიოდში იყიდა ბილეთი სისტემამ ამ კონკრეტულ ივენთზე განსაზღვრული ბილეთების რაოდენობას უნდა დააკლოს 1.

1.5.2 ივენთის დეტალების გვერდზე თავიდანვე ჩანს „დაჯავშნა“ და „ყიდვა“ ლილაკები. (მხოლოდ იმ შემთხვევაში თუ მომხმარებელი ავტორიზებულია). ლილაკები უნდა ჩანდეს იმის მიხედვით არის თუ არა აქტიური ბილეთები დარჩენილი.

„ყიდვა“ ლილაკზე დაჭერისას არსებული ბილეთების რაოდენობას აკლდება 1.

1.5.3 ღონისძიების დასრულების შემდეგ სისტემა ავტომატურად უნდა შლიდეს მთავარი გვერდიდან ამ კონკრეტულ ივენთს და გადაიტანოს არქივში (worker ის საშუალებით)

1.6 თითოეულ ივენთს ასევე უნდა გააჩნდეს რედაქტირების ფუნქცია, რომელზე დაჭერისას სისტემა გადავიდეს ახალ ფორმაზე და არსებული ინფორმაციით შეავსოს ფორმა , მომხმარებელს უნდა შეეძლოს შეცვალოს ივენთის დეტალები თუმცა ცვლილება შესაძლებელი უნდა იყოს ივენთის შექმნიდან X (დღის განმავლობაში) X - დღეს განსაზღვრავს ადმინისტრატორი თავისსავე პანელში

1.7 მომხმარებელს არ უნდა შეეძლოს თავისი შექმნილი ივენთის წაშლა.

## 2. ადმინისტრატორის პანელი (MVC)

1. 1 MVC აპლიკაციას უნდა გააჩნდეს ავტორიზაციის გვერდი (იუზერები შეიქმნება ბაზიდან ერთჯერადად)

1. 2 ადმინისტრატორს (როლი, რომელიც ადმინისტრატორის პანელს მართავს უნდა შეეძლოს აპლიკაციის იუზერების რედაქტირება(მიანიჭოს მოდერატორის სტატუსი - ექნება უფლება ივენთის დადასტურების), წაშლა, ხედვა. მოდერატორი - მხოლოდ event-ები, admin-event, ადმინისტრირება.

1.3 ადმინისტრატორს Event- ების მოდულში უნდა უჩანდეს ყველა ის ივენთი, რომელიც დაემატება მომხმარებლების მიერ საიტზე, მან უნდა გადანაცვლოს რომელი ივენთ დაიდება საიტზე - „გამოაჩინე“ (IsActive) ლილაკზე დაჭერით (რომელიც თითოეულ ივენთს ექნება)

1.4 აქვს გვერდი სადაც შეყავს კონკრეტული რიცხვი ივენთის ცვლილების ზღვარი (რაც გულისხმობს ივენთის დადებიდან X პერიოდს, რის შემდეგადაც ივენთის დეტალების ცვლილება აღარ იქნება შესაძლებელი)

### 3. 1. Events.ItAcademy.Ge – API აპლიკაცია

MVC ის ანალოგი პროექტი რომელიც ინტერფეისად იყენებს api controller ებს და swagger -ს. MVC აპლიკაცია და API იც ორივე იყენებს ერთსა და იმავე source code -ს \*არ დაადუბლიროთ, ანუ სოლუშენში გქონდეთ ორი Presentation პროექტი ერთი MVC მეორე API ორივე მათგანი მიმართავ Application - ს)

### აუცილებელი მოთხოვნები :

1. მონაცემთა ბაზა SQL server
2. ORM – Entity Framework Core Code First, (Seed, Migration)
3. Web Api Core, MVC Core
4. Api Versioning
5. Middllwares
6. Global Exception Handling
7. Asp Authorization

- 8. Swagger
- 9. Fluent Api
- 10. Mapster ან Mapper
- 11. DTO, VlewModel
- 12. Layered Architecture

#### სასურველი მოთხოვნები:

- 1. Paging