



გიორგი მეცხვარიშვილი

ძიუდოს შეჯიბრის ტაბლოს პროგრამული უზრუნველყოფა

ტექნიკური დავალება 1.0

GIORGI METSKHVARISHVILI



სარჩევი

პროექტის აღწერა	2
ტექნიკური პარამეტრები და მახასიათებლები	3
შეწყვილების ალგორითმი	3
მესამე ადგილის შეწყვილების ალგორითმი	3
მთავარი გვერდი	4
რეგიონების გვერდი	4
სპორტსმენების გვერდი	5
ტურნირების გვერდი	6
ტურნირზე მონაწილე სპორტსმენების გვერდი	7
ტურნირის შემადგენელი შეჯიბრებების გვერდი	8
შეჯიბრის კონტროლის გვერდი	9
შეჯიბრის გვერდი	10
მომხმარებლების გვერდი	10

პროექტის აღწერა

შესავალი

აპლიკაცია გათვლილია საქართველოს მიუდოს ფედერაციის მიერ ეროვნულ დონეზე ჩატარებულ შეჯიბრებზე, მისი მიმდინარეობისას საინფორმაციო ტაბლოს მართვასა და ინფორმაციული სახით ჩვენებაზე. ის ასევე აერთიანებს საქართველოში მოასპარეზე სპორტსმენებისა და ორგანიზებული ტურნირების ბაზას. აპლიკაცია შეიცავს ალგორითმს, რომლის დახმარებითაც შესაძლებელია ტურნირში მონაწილე სპორტსმენების შეწყვილების ცხრილის შედგენა, მოდიფიკაცია. აგრეთვე, მასში განთავსებულია მესამე ადგილის შეწყვილების ალგორითმი.

პროგრამის მოხმარება შეუძლია არაავტორიზებულ მომხმარებელს საინფორმაციო მიზნებით და ავტორიზებულ მომხმარებელს სხვადასხვა დონეზე სამართავად. აპლიკაცია გამიზნულია live რეჟიმზე. აქედან გამომდინარე, ნებისმიერ მომხმარებელს შეუძლია თვალი ადევნოს ტურნირის მსვლელობას დისტანციურად, ხოლო ტურნირზე დამსწრე მაყურებელს - ტაბლოს სახით.

ძირითადი ამოცანები

- სისტემის ორგანიზება:
სისტემაში უნდა არსებობდეს მონაცემთა ბაზა, სადაც შენახული იქნება ინფორმაცია სპორტსმენებისა და ტურნირების შესახებ;
- ტურნირის შედგენის ამოცანა:
საწყის საფეხურზე იქმნება ტურნირი, შემდგომ აღნიშნულ შეჯიბრსვე ემატება მონაწილეები. შედეგის მიხედვით იქმნება შეწყვილების ცხრილი, რომლის მიხედვითაც ტარდება ტურნირი.
- მესამე ადგილის შეწყვილების ცხრილის შედგენის ამოცანა
დამუშავების პროცესშია;
- ტაბლოს მართვა:
ტაბლოს მართვა დაფუძნებულია მიუდოს შეჯიბრის პრინციპზე. შეჯიბრი მიმდინარეობს ორ სპორტსმენს შორის ძირითადი 4 წუთის განმავლობაში. თუ ძირითადი 4 წუთის ამოწურვის შემდგომ ვერ გამოვლინდა გამარჯვებული, ათვლას დაიწყებს დამატებითი დრო გამარჯვებულის გამოვლენამდე. გამარჯვება განისაზღვრება ორი ტიპის ქულათა სისტემით. 1. იპონი - მისი მნიშვნელობა შეიძლება იყოს 1 ან 0. თუ რომელიმე სპორტსმენი გააკეთებს იპონს შეჯიბრი სრულდება და გამარჯვებულად ცხადდება ზემოაღნიშნული მოჭიდავე. 2. ვაზარი - მისი მნიშვნელობა შეიძლება იყოს 0 ან ნებისმიერი ნატურალური რიცხვი. თითო ილეთის გაკეთება აისახება 1 ქულის მომატების სახით. გამარჯვებულია ის მონაწილე რომელიც მეტ ვაზარს ქულას დააგროვებს. ამას დამატებით ჯარიმის სახით თითოეულმა მონაწილემ შეიძლება მიიღოს მაქსიმუმ 3 გაფრთხილება. მესამე გაფრთხილებისას მონაწილე ეთიშება შეჯიბრს და გამარჯვებულად ცხადდება მისი



ოპონენტი. მონაწილე შეიძლება ასევე დისკვალიფიცირდეს ტურნირის
მიმდინარეობისას წონითი კატეგორიის დარღვევის გამო. ამ შემთხვევაშიც,
გამარჯვებულად ცხადდება მისი ოპონენტი სპორტსმენი.

ტექნიკური პარამეტრები და მახასიათებლები

- Programming Language: C#;
- Web Framework: ASP.NET MVC 5;
- Software Architecture: N-tier (Multilayered) Architecture;
- Design-Patterns: Inversion of Control, Dependency Injection;
- Dependency Principal (Design-Patterns Implementation) : Ninject;
- Database: MSSQL Server;
- Data Access Technology: Entity Framework;
- UI Library: Telerik Kendo UI MVC;
- Real-time Web Functionality => (Sockets): ASP.NET SignalR;
- Helper Library for File Generations: DevExpress;
- UI Theme Library: Keen Themes - Metrinic;
- Front-End Languages: HTML, CSS, Javascript, JQuery;
- Front-End Component Library: Bootstrap.

შეწყვილების ალგორითმი

ალგორითმი დამუშავების პროცესშია;

მესამე ადგილის შეწყვილების ალგორითმი

ალგორითმი დამუშავების პროცესშია;



მთავარი გვერდი

აღწერა:

მთავარ გვერდს უნდა ჰქონდეს ვიზუალური დატვირთვა. მასზე უნდა განთავსდეს ანიმაციური სლაიდერი, რომელზეც გამოსახული იქნება მიუდოსთან დაკავშირებული სურათები.

უფლებები:

გვერდზე დაიშვებიან ავტორიზებული და არაავტორიზებული მომხმარებლები;

ხილვადობა თავსართში:

- SuperAdmin: მთავარი, ტურნირები, სპორტსმენები, რეგიონები, მომხმარებლები;
- Admin: მთავარი, ტურნირები, სპორტსმენები, რეგიონები, მომხმარებლები;
- User: მთავარი, ტურნირები, სპორტსმენები;
- არაავტორიზებული მომხმარებელი: მთავარი, ტურნირები, სპორტსმენები.

დამატებითი ინფორმაცია:

თავსართის ზემოთ განთავსებული უნდა იყოს ლოგო, დასახელება. იმ შემთხვევაში, თუ მომხმარებელი ავტორიზებულია - მომხმარებელიდან გამოსვლა და პაროლის შეცვლა.

რეგიონების გვერდი

აღწერა:

რეგიონების გვერდზე განთავსებული უნდა იყოს საქართველოში არსებული ყველა ქალაქის ჩამონათვალი, რომლებსაც შემდგომში სპორტსმენების სისტემაში დამატებისას მივუთითებთ.

ფუნქციონალი:

- რეგიონის დამატება;
- არსებული რეგიონის ჩასწორება;
- არსებული რეგიონის წაშლა.

უფლებები:

გვერდზე შესვლა და ფუნქციონალის გამოყენება შეუძლია მხოლოდ SuperAdmin-სა და Admin-ს.

ვიზუალიზაცია:

ინფორმაციის გამოსატანად და მათ დასამუშავებლად გამოყენებული იქნება Telerik UI Grid Inline Editing - ის ფუნქციონალით.



ვალიდაცია:

Client Side ვალიდაციისთვის გამოყენებული იქნება Remote Validation, რომელიც შეამოწმებს რომ არ მოხდეს ქალაქის სახელის დუბლირება.

ლოგირება:

Action Log ცხრილში შეინახოს ყველა POST ტიპის მეთოდი (Add, Edit, Delete).

ბაზის მოდელი:

- იდენტიფიკატორი: int Id [Key]
 - დასახელება: string Name
- ყველა ველი სავალდებულოა

სპორტსმენების გვერდი

აღწერა:

სპორტსმენების გვერდზე განთავსებული უნდა იყოს საქართველოში მოასპარეზე ყველა სპორტსმენის ჩამონათვალი, რომლებსაც შემდგომში ტურნირების სისტემაში სპორტსმენების დამატებისას მივუთითებთ.

ფუნქციონალი:

- სპორტსმენის დამატება;
- არსებული სპორტსმენის ჩასწორება;
- არსებული სპორტსმენის წაშლა.

უფლებები:

გვერდზე დაიშვებიან ავტორიზებული და არაავტორიზებული მომხმარებლები; ფუნქციონალის გამოყენება შეუძლია მხოლოდ SuperAdmin-სა და Admin-ს.

ვიზუალიზაცია:

ინფორმაციის გამოსატანად და მათ დასამუშავებლად გამოყენებული იქნება Telerik UI Grid Window Editing - ის ფუნქციონალით.

ლოგირება:

Action Log ცხრილში შეინახოს ყველა POST ტიპის მეთოდი (Add, Edit, Delete).

ბაზის მოდელი:

- იდენტიფიკატორი: int Id [Key]
- სახელი: string Name
- გვარი: string Surname
- წონა: double Weight
- რეიტინგი: int Rank



- რეგიონი: RegionId
ყველა ველი სავალდებულოა

ტურნირების გვერდი

აღწერა:

ტურნირების გვერდზე განთავსებული უნდა იყოს საქართველოს ძიუდოს ფედერაციის მიერ საქართველოში ორგანიზებული ყველა ტურნირის ჩამონათვალი. ტაბების სახით უნდა გამოიყოს მიმდინარე, მომავალი და უკვე ჩატარებული ტურნირები. ადმინისტრატორს უნდა შეემლოს ტურნირის მართვა (სპორტსმენების დამატება და შეწყვილების ცხრილის აგება), User-ს და არაავტორიზებულ მომხმარებელს მიმდინარე და მომავალი ტურნირის დეტალების(ინფორმაციის და შედგენილი შეწყვილების ცხრილის) დათვალიერება.

ფუნქციონალი:

- ტურნირის დამატება;
- არსებული ტურნირის ჩასწორება;
- არსებული ტურნირის წაშლა;
- არსებული ტურნირის მართვა;
- არსებული ტურნირის დეტალური ინფორმაციის ნახვა.

უფლებები:

გვერდზე დაიშვებიან ავტორიზებული და არაავტორიზებული მომხმარებლები;

- SuperAdmin: მიმდინარე (შესაძლებელია მოდიფიკაცია) , მომავალი (შესაძლებელია მოდიფიკაცია) და უკვე ჩატარებული ტურნირები;
- Admin: მიმდინარე (შესაძლებელია მოდიფიკაცია) , მომავალი (შესაძლებელია მოდიფიკაცია) და უკვე ჩატარებული ტურნირები;
- User: მიმდინარე და მომავალი ტურნირები;
- არაავტორიზებული მომხმარებელი: მიმდინარე და მომავალი ტურნირები.

ვიზუალიზაცია:

ინფორმაციის გამოსატანად გამოყენებული იქნება 3 tab-ი (მომდინარე, მომავალი და უკვე ჩატარებული ტურნირები)-თვის. თითოეულისთვის გამოჩნდება შესაბამისი კონტენტი ჩამონათვალის სახით. მომხმარებლისთვის, რომლისთვისაც დაშვებულია მოდიფიკაცია, ტურნირის single layout-ზე უნდა იყოს განთავსებული მართვის ღილაკი. თითოეულ მათგანზე უნდა გამოჩნდეს ტურნირის დასახელება, აღწერა, თარიღი, წონითი კატეგორია და ტურნირის მონაწილეთა მაქსიმალური რაოდენობა.

ვალიდაცია:

Client Side ვალიდაციისთვის გამოყენებული იქნება Remote Validation, რაც შეამოწმებს რომ არ მოხდეს ტურნირის სახელის დუბლირება.

ლოგირება:

Action Log ცხრილში შეინახოს ყველა POST ტიპის მეთოდი (Add, Edit, Delete).

ბაზის მოდელი:

- იდენტიფიკატორი: int Id [Key]
 - დასახელება: string Name
 - აღწერა: string Description
 - მონაწილეთა მაქსიმალური რაოდენობა: int NumberOfContestants
 - თარიღი: DateTime Date
 - წონოთი კატეგორია: int Weight
 - სტატუსი: int StatusId
- ყველა ველი სავალდებულოა

ტურნირზე მონაწილე სპორტსმენების გვერდი

აღწერა:

ტურნირზე მონაწილე სპორტსმენების გვერდზე გადასვლა მოხდება ტურნირების გვერდიდან კონკრეტული ტურნირზე განთავსებული მართვის ლილაკზე დაჭერით. გვერდზე განთავსებული იქნება აღნიშნულ ტურნირზე მონაწილე ყველა სპორტსმენის ჩამონათვალი და შესაძლებელი იქნება მათი მოდიფიკაცია ტურნირის დაწყებამდე. სპორტსმენების დამატების შემდგომ შესაძლებელი იქნება ტურნირის შედგენა ლილაკზე დაჭერით, რითიც მოხდება სპორტსმენების დაწყვილება და შეწყვილების ცხრილის შედგენა სპეციალური ალგორითმის მეშვეობით.

ფუნქციონალი:

- არსებულ ტურნირზე სპორტსმენის დამატება, განთესვა;
- არსებულ ტურნირზე სპორტსმენის წაშლა;
- არსებული ტურნირის მართვა;
- არსებული ტურნირის დეტალური ინფორმაციის ნახვა;
- არსებული ტურნირის შეწყვილების ცხრილის აგება.

უფლებები:

გვერდზე შესვლა და ფუნქციონალის გამოყენება შეუძლია მხოლოდ SuperAdmin-სა და Admin-ს.

ვიზუალიზაცია:

ინფორმაციის გამოსატანად და მათ დასამუშავებლად გამოყენებული იქნება Telerik UI Grid Window Editing - ის ფუნქციონალით.

ვალიდაცია:

Client Side ვალიდაციისთვის გამოყენებული იქნება Remote Validation, რაც შეამოწმებს რომ არ მოხდეს ტურნირზე სპორტსმენის დუბლირება.

ლოგირება:

Action Log ცხრილში შეინახოს ყველა POST ტიპის მეთოდი (Add, Edit, Delete).

ბაზის მოდელი:

- იდენტიფიკატორი: int Id [Key]
 - ტურნირის იდენტიფიკატორი: int TournamentId
 - სპორტსმენის იდენტიფიკატორი: int SportsmanId
 - განთესვა: bool Seeded
- ყველა ველი სავალდებულოა

ტურნირის შემადგენელი შეჯიბრებების გვერდი

აღწერა:

ტურნირის შემადგენელი შეჯიბრებების გვერდზე განთავსებულია დეტალური ინფორმაცია ტურნირის შესახებ და ასევე შეწყვილების ცხრილი, თუ ის უკვე შედგენილია. მასზე გამოსვლა ხდება ტურნირების გვერდიდან კონკრეტული ტურნირზე განთავსებული დეტალების ღილაკზე დაჭერით. იმ შემთხვევაში, თუ ტურნირი ჯერ არ დაწყებულა ადმინისტრატორის მიერ შესაძლებელია შეწყვილების ცხრილის შედგენა ტურნირზე მონაწილე სპორტსმენების გვერდზე „ტურნირის შედგენის“ ღილაკზე დაჭერით. ასევე აღნიშნულ გვერდზე შესაძლებელი უნდა იყოს სპორტსმენების გადაადგილება შეწყვილების ცხრილში.

ფუნქციონალი:

- არსებულ შეწყვილების ცხრილში სპორტსმენების გადაადგილება.

უფლებები:

გვერდზე დაიშვებიან ავტორიზებული და არაავტორიზებული მომხმარებლები; ფუნქციონალის გამოყენება შეუძლია მხოლოდ SuperAdmin-სა და Admin-ს.

ვიზუალიზაცია:

ინფორმაციის სახით უნდა გამოჩნდეს ტურნირის შესახებ დეტალური ინფორმაცია და აიგოს შეწყვილების ცხრილის ვიზუალური მხარე.

ლოგირება:

Action Log ცხრილში შეინახოს ყველა POST ტიპის მეთოდი.

ბაზის მოდელი:

- იდენტიფიკატორი: int Id [Key]
 - ტურნირის იდენტიფიკატორი: int TournamentId
 - შეჯიბრის იდენტიფიკატორი: int WrestleId
- ყველა ველი სავალდებულოა

შეჯიბრის კონტროლის გვერდი

აღწერა:

შეჯიბრის კონტროლის გვერდზე განთავსებულია ორ სპორტსმენს შორის შეჯიბრის მართვის ფუნქციონალი და ინფორმაციის გამოტანა.

ფუნქციონალი:

- სპორტსმენებისთვის იპონის ქულის დამატება/დაკლება/გამოჩენა;
- სპორტსმენებისთვის ვაზარის ქულის დამატება/დაკლება/გამოჩენა;
- სპორტსმენებისთვის გაფრთხილების ქულის დამატება/დაკლება/გამოჩენა;
- შეჯიბრის დროის დაწყება, გაჩერება, განულება;
- მოქმედების ტექსტის დამატება/წაშლა/გამოჩენა;
- მოქმედების ტექსტის ფაილად გენერირება.

უფლებები:

გვერდზე შესვლა და ფუნქციონალის გამოყენება შეუძლია მხოლოდ ავტორიზებულ მომხმარებელს.

ლოგირება:

Action Log ცხრილში შეინახოს ყველა POST ტიპის მეთოდი.

დამატებითი ინფორმაცია:

ფუნქციონალის ქმედებების შესრულებისას ინფორმაცია უნდა შეინახოს ბაზაში და აისახოს როგორც შეჯიბრის კონტროლის გვერდზე ასევე შეჯიბრის გვერდზე, SignalR - ის გამოყენებით.

ბაზის მოდელი:

- იდენტიფიკატორი: int Id [Key]
- პირველი სპორტსმენის იდენტიფიკატორი: int SportsmanOneId
- მეორე სპორტსმენის ერთი იდენტიფიკატორი: int SportsmanTwoId
- პირველი სპორტსმენის ვაზარის ქულა: int SportsmanOneWazariPoint
- მეორე სპორტსმენის ვაზარის ქულა: int SportsmanTwoWazariPoint
- პირველი სპორტსმენის იპონის ქულა: bool SportsmanOneIppon
- მეორე სპორტსმენის იპონის ქულა: bool SportsmanTwoIppon



- პირველი სპორტსმენის ჯარიმის ქულა: int SportmanOneFine
- მეორე სპორტსმენის ჯარიმის ქულა: int SportmanTwoFine
- შეჯიბრის დრო: Time time
- დროის მიმართულება: bool Decreasing
- სტატუსი: int statusId
- გამარჯვებული სპორტსმენის იდენტიფიკატორი: int WinnerId
ყველა ველი სავალდებულოა

შეჯიბრის გვერდი

აღწერა:

შეჯიბრის გვერდზე განთავსებულია ორ სპორტსმენს შორის შეჯიბრის ინფორმაციის გამოტანა.

ფუნქციონალი:

- ინფორმაციის გამოჩენა

უფლებები:

გვერდზე დაიშვებიან ავტორიზებული და არაავტორიზებული მომხმარებლები;

მომხმარებლების გვერდი

აღწერა:

მომხმარებლების გვერდზე უნდა განთავსდეს სისტემაში არსებული ყველა მომხმარებლის ჩამონათვალი.

ფუნქციონალი:

- მომხმარებლის დამატება;
- არსებული მომხმარებლის რედაქტირება;
- არსებული მომხმარებლის წაშლა;

უფლებები:

გვერდზე შესვლა და ფუნქციონალის გამოყენება შეუძლია მხოლოდ Administrator მომხმარებელს.

- SuperAdmin: ყველა ტიპის მომხმარებელზე მოდიფიკაცია;
- Admin: User ტიპის მომხმარებლებზე მოდიფიკაცია;



ვიზუალიზაცია:

ინფორმაციის გამოსატანად და მათ დასამუშავებლად გამოყენებული იქნება Telerik UI Grid Window Editing - ის ფუნქციონალით.

ვალიდაცია:

Client Side ვალიდაციისთვის გამოყენებული იქნება Remote Validation, რომელიც შეამოწმებს რომ არ მოხდეს მომხმარებლების დუბლირება.

ლოგირება:

Action Log ცხრილში შეინახოს ყველა POST ტიპის მეთოდი (Add, Edit, Delete).

ბაზის მოდელი:

- იდენტიფიკატორი: string Id [Key]
 - სახელი: string FirstName
 - გვარი: string LastName
 - მომხმარებლის სახელი: string UserName
 - პაროლი: string PasswordHash
 - სტატუსი: bool IsDeleted
- ყველა ველი სავალდებულოა