

ივანე ჯავახიშვილის სახელობის თბილისის
სახელმწიფო უნივერსტიტეტი

ვებ ტექნოლოგიები სამეცნიერო კონფერენციის
მენეჯმენტისთვის

პროექტის ავტორები :

თენგიზ ბრეგვაძე

ირაკლი გუგუნაშვილი

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი

პროექტის ხელმძღვანელი:

ასოცირებული პროფესორი მაგდა ცინცაძე

ქალაქი თბილისი 2018 წელი

სარჩევი

ანოტაცია	გვ 3
შესავალი	გვ 4
გამოყენებული ტექნოლოგიები	გვ 5
ვებ-გვერდის ფუნქციონალი	გვ 7
დასკვნა	გვ 9
გამოყენებული ლიტერატურა	გვ 9
დანართები	გვ 10

ანოტაცია

ჩვენს მიერ წარმოდგენილი რეფერატი აღწერს თუ რა პროცესს გადის ვებ-აპლიკაცია პროექტის აღებიდან მის დასრულებამდე. ჩვენ თბილისის სახელმწიფო უნივერსიტეტის კომპიუტერული მეცნიერებების კომფერენციის ვებ-გვერდის შექმნის მაგალითზე განვიხილავთ ჩვენს შეხედულებებს და მათ იმპლიმენტაციას თუ რა სახის სამუშაო არის გასაწევი ასეთი ტიპის პროექტზე, რომელი toolkit არის ოპტიმალური დეველოპერებისთვის ასეთი კალიბრის პროექტისათვის საჭირო სოფტის შესაქმნელად და გასატესტად, როგორ ხდება პროექტის მოთხოვნების მიხედვით ტექნიკური დავალების შედგენა ანალიზის ფაზაში ანსაჭიროების შემთხვევაში რა კომპრომისებზე წასვლაა საჭირო. ასევე განვიხილავთ და ავლწერთ რა ფაქტორები ახდენენ ზეგავლენას და რამდენად თავისუფლები ვართ არქიტექტურული დიზაინის არჩევისას, ასევე რამდენად გვზღუდავს მოცემული პროექტი არჩევანის გაკეთების პროცესში. მომდევნო პუნქტებში, ჩვენი პროექტის მაგალითზე წარმოვადენთ ჩვენს მიზეზებს და შეხედულებებს თუ რაზე შევაჩერეთ არჩევანი ფრონტ ენდის, ბექისთვის და როგორ მოვახდინეთ მონაცემთა ბაზის იმპლემენტირება. მიმოვიხილავთ კოდის დაწერის პროცესს, სხვადასხვა კომპონენტების დაწერის თანმიმდევრობას, რა ტიპიურ პრობლემებს ვაწყდებით ხოლმე ამ ფაზაში და მათი გადაჭრის ჩვენეულ გზებს. ასევე ვაჩვენებთ თუ რა გზებით იქნება შესაძლებელი მოგვიანებით ვებ-გვერდის გავრცობა მას შემდეგ რაც ჩვენ დავასრულებთ მასზე მუშაობას.

This thesis presented by us describes web-application and stages it undergoes from the beginning of the project to its completion. We, using the example of building conference web-page for Tbilisi state university, will present our views and their implementations on how such project must be undergone, what type of toolkit is optimal for developers when creating and testing the software for the project of such caliber. How we determine the course of action according to the requirements of the project in the analysis phase, or what sort of compromises are necessary depending on the situation. We will also discuss and describe factors that determine degree of freedom we have when choosing architectural design, and how specifics of given project can limit us when making the choice. In the following paragraphs, using the example of our project, we will present our reasons and views why and what choices we made for implementing front and back ends of the web-page and how database is structured. We will discuss process of writing a code, orders in which we create various components, common difficulties we encounter in this phase and our ways of solving them. We will also demonstrate the ways in which it will be possible to add on additional features to the web-page after project has been completed.

შესავალი

სანამ დავიწყებდეთ პროექტის შესრულებას, ამუცილებელია გავიაზროთ თუ რა ტექნიკურ დავალებასთან გვაქვს საქმე, რა ფუნქციონალი უნდა ჰქონდეს ვებ-გვერდს და რა სტანდარტებს უნდა აკმაყოფილებდეს შესრულებული პროექტი, რა ტექნოლოგიებს გამოვიყენებთ ამოცანის გადასაჭრელად და რა სახის არქიტექტურას შევარჩევთ ჩვენი ვებ-პორტალისთვის.

თავდაპირველად მიმოვიხილოთ ვებ-გვერდის ფუნქციონალი, რადგან სწორედ მისი იმპლემენტაციის დროს განვლილი გზის აღწერაა ჩვენი პროექტის ერთ-ერთი ძირითადი მიზანი. რადგან ვებ-პორტალის მიზანია მენეჯმენტი გაუწიოს სამეცნიერო კომფერენციას, მისი ფუნქციონალის უპირველესი ნაწილი უნდა იყოს ინფორმაციის მართვა. სწორედ ამ სერვისის საშუალებით უნდა იყოს შესაძლებელი ყველა დაინტერესებულ მხარეს მიეწოდოს ყველა ის აუცილებელი ინფორმაცია რაც კონფერენციის მსვლელობის დროს გახდება აქტუალური, მაგალითად: კონფერენციის ფარგლებში გამართული ღონისძიებების კალენდარი, დოკუმენტაცია სამეცნიერო ნაშრომებისა და მოხსენებების სტანდარტებისა და კრიტერიუმების შესახებ, სიახლეების ფორუმი, კომფერენციის ადმინისტრაციის დადგენილებათა და განცხადებათა საინფორმაციო დაფა და ასე შემდეგ. ვებ-პორტალის ამ ნაწილთან გვერდის ყველა სტუმარს უნდა ჰქონდეს წვდომა, რადგან აქ ხდება საჯარო ინფორმაციის განთავსება. სამეცნიერო კომფერენციის არსი სამეცნიერო საქმიანობის წარმოჩენაა, ამიტომაც ვებ-გვერდზე უნდა შეგვეძლოს კომფერენციის დარეგისტრირებულ მონაწილეთა ნაშრომების, მოხსენებების, კვლევების ატვირტვა და გამოქვეყნება, რომელიც ლოგიკურად დაჯგუფდება კატეგორიების მიხედვით, რათა მომხმარებელთა მიერ გაადვილდეს საჭირო ნაშრომების მოძიება და მათზე წვდომა. ყოველ ასეთ ღონისძიებას ყავს სპონსორები, პარტნიორები და მოწვეული სტუმრები, ამიტომ ვებ-პორტალზე მიზანშეწონილია რომ განთავსებული იყოს მათ შესახებ ნიფორმაცია, თუ ვინ არიან ისინი, რა მიზნები ამოძრავებთ, რა სფეროებში მოღვაწეობენ და ასე შემდეგ, ამგვარად კომფერენციით დაინტერესებული ყველა ის პირი ვინც ეწვევა ვებ-გვერდს შეძლებს განსაზღვროს თუ რა დატვირთვა აქვს ღონისძიებას, დაინტერესდეს პარტნიორი ან სპონსორი კომპანიის საქმიანობით, ან შეიტყოს კონფერენციაზე მოწვეული სპეციალისტების შესახებ. რადგან სამეცნიერო კომფერენცია საჯარო ღონისძიებაა, აუცილებელია განვათავსოთ საკონტაქტო ინფორმაცია ადმინისტრაციასთან, მისამართები, რათა დაინტერესებულმა პირებმა გაიადვილონ კომფერენციაში მონაწილეობის მიღება.

აქედან გამომდინარე ჩვენ უკვე შეგვიძლია შევიქმნათ ზოგადი წარმოდგენა თუ რა სახე ექნება ვებ-გვერდს, მაგრამ ამ ეტაპზე პირდაპირ კოდის წერა არ იქნება გამართლებული, ჯერ უნდა გავითვალისწინოთ ძირითადი პრინციპები რომლებიც ახასიათებს კარგად აწყობილ ვებ გვერდს და აკმაყოფილებს ინდუსტრიის მიმდინარე ტერქნოლოგიურ სტანდარტებს.

თავდაპირველად იმის გათვალისწინებით, რომ ეს არის ჯგუფური პროექტი, კარგი იქნება თუ განვაცალკევებთ ბიზნეს ლოგიკასა და პრეზენტაციის ლოგიკას, გარდა იმისა რომ ეს საზოგადოდ კარგ სტანდარტად ითვლება, ასე გაცოფილ პროექტში საქმის გადანაწილება დეველოპერების მიერ უფრო მარტივად ხდება. თან ასეთი მიდგომით ფრონტ-ენდ დეველოპერებს ექნებათ საშუალება მოახდინონ კონცენტრაცია პრეზენტაციის ლოგიკაზე, ისე რომ შეხება იქონიონ ბექ-ენდთან ნეთვორქინგის ან ზოგად არიტექტურასთან დაკავშირებულ საკითხებში. დიდ პროექტებში კი ეს მიდგომა აუცილებელიც კი არის შესასრულებელი სამუშაოს მოცულობის გამო. ეს კოდს ადვილად წასაკითხსაც ხდის და საშვალეებს იძლევა რომ დეველოპერებმა პარალელურად იმუშავონ რამოდენიმე კომპონენტზე. ასევე გადასაჭრელი გვაქვს თანამედროვე ვებ გვერდების ერთ-ერთი მთავარი გამოწვევა, რომელიც არის საიტის დიზაინის ისეთნაირად შექმნა, რომელიც უზრუნველყოფს იმას, რომ როდესაც მომხმარებელი ეწვევა ვებ გვერდს, ის შემდგომში მიიღოს ინფორმაცია ადვილად აღსაქმელ ფორმატში. ამას ართულებს ის ფაქტი რომ მომხმარებლები სტუმრობენ გვერდს სხვადასხვა ელექტრონული მოწყობილობებით განსხვავებული ეკრანის ზომით და რეზოლუციით. ასევე გასათვალისწინებელია ის რომ ვებ გვერდი სწორედ უნდა წარმოჩინდეს სხვადასხვა ბრაუზერში, სხვადასხვა ოპერატიულ სისტემებზე და რაგორც უკვე ავლინდნით, სხვადასხვა ელექტრონულ მოწყობილობებზე, ამისთვის კი ვებ გვერდის დიზაინის სპეციფიკურად დაგეგმვა საჭირო დაგეგმვა საჭირო. რაც შეეხება ტექნოლოგიებს, არჩევანი შევაჩერეთ ASP.Net-ზე კონკრეტულად კი ASP.Net MVC 5-ზე, რადგან ეს არის ძლიერი ხელსაწყო web აპლიკაციების შესაქმნელად, რომელსაც შეუძლია დინამური ვებ გვერდების დაგენერირება რამდენიმე იყენებენ HTML, CSS, JavaScript-ს

გამოყენებული ტექნოლოგიები

ჩვენს მიერ შესრულებული სამეცნიერო კომფერენციის ვებ-გვერდი სრულად არის შესრულებული ASP.Net MVC 5-ზე, თვითონ MVC(Model-View-Controller) არის არქიტექტურული პატერნი რომელიც გამოიყენება ვებ აპლიკაციების შესაქმნელად და რომელიც აპლიკაციას 3 ერთმანეთთან დაკავშირებულ ნაწილად ყოფს.

მისი მუშაობის პრინციპი ასეთია: მომხმარებელი ხედავს view-ს, იყენებს controller-ს, ამ დროს controller-ი მანიპულაციას ახდენს model-ზე, ხოლო შეცვლილი model ანახლებს view-ს. რადგან MVC-ს დიზაინი აცალკევებს ერთმანეთისგან ან ძირითად კომპონენტებს, ეს საშუალებას იძლევა გამოვიყენოთ კოდის ბლოკები და ამით შევამციროთ დეველოპმენტზე დახარჯული რესურსი, ხოლო იმის საშუალება რომ პარალელურად ვიმუშავოთ კომპონენტებზე გვადლევს საშუალებას ოპტიმალურად გამოვიყენოთ დეველოპმენტისთვის განკუთვნილი დრო.

ასევე რადგან ჩვენ სრულიად თავიდან ვიწყებთ საიტის აწყობას და არ გვიწევს რაიმე არსებული პროექტის გადაკეთება ASP.Net MVC ყველაზე კარგი არჩევანია, რადგან აქ გაცილებით გამარტივებულია ტესტირების საკითხი, რაც ძალიან დიდი უპირატესობაა დიდი და საშუალო ზომის პროექტებისთვის რაც დეზაგინგის და უნიტ ტესტირებისთვის გაცილებით ნაკლები დროის დახარჯვას გულისხმობს.

რაც შეეხება ფრონტ ენდს, კლიენტის მხარეს მას ჩვენ ვქმნით MVC 5-ის view კომპონენტში, ფრონტ ენდ ვებ დეველოპმენტის ძირითადი პრინციპი ძალიან მარტივი ასაღწერია, ეს არის სამი ფუნდამენტური ტექნოლოგიის რომელიც გამოიყენება ინტერნეტ პროდუქციის შესაქმნელად: HTML, CSS და JavaScript-ის საშუალებით ვებ-გვერდის ან ვებ აპლიკაციის შექმნა ისე, რომ მომხმარებელს საკუთარი ელექტრონული მოწყობილობის საშუალებით უშუალოდ შეეძლოს მისი ნახვა და დანიშნულებისამებრ გამოყენება. ფრონტ ენდში ჩვენ ასევე ვიყენებთ Razor-ს.

Razor არის მარკაპ სინტაქსი, რომელიც საშუალებას გვაძლევს ჩავაშენოთ სერვერზე დაფუძნებული კოდი უშუალოდ ვებ-გვერდებში. აღსანიშნავია რომ Razor არ არის პროგრამირება ენა, ეს არის სერვერის მხარის მარკაპის ენა. Razor-ის სინტაქსი არის HTML, C# და Razor-ის მარკაპის ვარიაცია, ხოლო მისი ფაილური გაფართოვებაა .cshtml. Razor მარკაპიდან Html-ის დარენდერება ზუსტად იგივე პრინციპით ხდება როგორც Html ფაილიდან.

ჩვენ ასევე ვიყენებთ CSS Flexbox-ს (CSS Flexible Box Layout Module) ეს არის CSS3-ში ახალი განლაგების მოდული რომელიც გამოიყენება ვიზუალურ მხარეში ობიექტების ერთმანეთის მიმართ სიმეტრიულად განლაგებაში განსაზღვრული მიმართულებით ან მიმდევრობით, მაშინაც კი როცა გვერდი დინამიურია. Flex კონტეინერის ძირითადი თვისებაა მისი შესაძლებლობა რომ შეცვალოს მისი ტიპის ობიექტების სიგრძე და სიგანე რათა მათ სიმეტრიულად გადაავსონ არსებული სივრცე სხვადასხვა ზომის და რეზოლუციის ეკრანებზე.

MVC 5-ის ცენტრალური კომპონენტია კონტროლერი, რადგან კონტროლერში ხდება მონაცემთა მართვა, აპლიკაციის წესებისა და ლოგიკის გაწერა. კონტროლერი არის აპლიკაციის დინამიური მონაცემთა სტრუქტურა, და სწორედ აქ ხდება აპლიკაციაში ინფორმაციის მართვა მომხმარებლის ინტერეაქციის საფუძველზე. მონაცემებთან მუშაობისას და მონაცემთა ბაზის გენერირებისთვის კი ვიყენებთ Entity Framework

Entity Framework-ი წარმოადგენს .NET ფრეიმვორკზე დამფუძნებულ ობიექტზე ორიენტირებულ ტექნოლოგიას მონაცემებთან სამუშაოდ. Entity Framework-ი გვაძლავს საშუალებას ადვილად დავაკონექტდეთ და ვიმუშაოთ მონაცემთა ბაზებთან, მაგალითად მონაცემების წაშლა, დამატება, ცვლილებების შეტანა, დასელექტება. ასევე ცხრილების შექმნა მონაცემთა ბაზაში, და მოკლედ ყველანაირი საჭირო მანიპულაციის გაკეთება. ნებისმიერ არსებას, ისევე როგორც ობიექტს აქვს თავისი თვისებები. მაგალითად ადამიანს

გამოვარჩევთ ისეთ თვისებებით: სახელი, გვარი, სიმაღლე, წონა. არ არის აუცილებელი თვისებას გამოხატავდეს ისეთი მონაცემი როგორც int-ი, არამეტ შეიძლება გამოხატოთ იგი მეტ კომპლექსურ სტრუქტურად. ნებისმიერ არსებას შეიძლება ჰქონდეს ერთი ან მეტი თვისება, რომლითაც შეგვიძლია გამოვარჩიოთ იგი სხვა არსებებისაგან. ეგეთ თვისებებს Key(გასაღები) ჰქვია. არსები შეიძლება დაკავშირებული იყონ სხვა არსებებთან: ერთი ბევრთან, ერთი ერთთან და ბევრი ბევრთან, როგორც რეალურ მონაცემთა ბაზაში ხდება კავშირი გარე Key-ების მეშვეობით. Entity Framework-ის ერთ-ერთი თვისებაა LINQ გამოძახების გამოყენება მონაცემთა ბაზიდან მონაცემების გამორჩევისთვის. აგრეთვე აქვს ეგეთი თვისება როგორც Entity Data Model. ეს მოდელი არიგებს არსებების კლასებს რეალურ ტაბულებთან მონაცემთა ბაზაში. Entity Data Model სამი დონისაგან შედგება: კონცეფტუალური, საცავის დონე და mapping-ის დონე. კონცეფტუალურ დონეზე ხდება აპლიკაციაში გამოყენებული კლასების გარჩევა. საცავის დონეზე ხდება ტაბულების და კლასების დამოკიდებულებების გარჩევა. Mapping-ის დონე არის წინა ორის შუამავალი.

Entity Framework-ი გვაძლევს მონაცემთა ბაზასთან ურთიერთქმედების სამ ხერხს: Database First: Entity Framework-ი ქმნის კლასების კრებულს, რომელიც ასახავს კონკრეტული მონაცემთა ბაზის მოდელს. Model First: პირველ რიგში დეველოპერი ქმნის მონაცემთა ბაზის მოდელს, შემდეგ კი Entity Framework-ი ქმნის რეალურ მონაცემთა ბაზას სერვერზე. Code First: დეველოპერი ქმნის მონაცემების მოდელის კლასს რომლებიც შეინახება მონაცემთა ბაზაში, შემდეგ კი Entity Framework-ი ამ მოდელის მიხედვით შექმნის მონაცემთა ბაზას და მის ტაბულებს. ჩვენც სწორედ ამ ხერხს ვიყენებთ. ასეთნაირად გენერირებული მონაცემთა ბაზის .mdf ფაილი ჩვენ შეგვიძლია პროექტიდან ამოვაექსპორტოთ და ნებისმიერ სხვა არქიტექტურაში გამოვიყენოთ ყოველგვარი დაბრკოლების გარეშე.

ვებ-გვერდის ფუნქციონალი

იმისათვის რომ სამეცნიერო კომფერენციის ვებ-გვერდი იყოს ფუნქციონალურად მოწოდების სიმაღლეზე, გარდა იმისა რომ გვერდი იყოს რესპონსივული და კარგი დიზაინის მქონე, ვიზუალურად აკმაყოფილებდეს თანამედროვე ვებ-გვერდების სტანდარტებს, მას უნდა ჰქონდეს მისთვის სპეციფიკური განსაკუთრებული ფუნქციონალი. ჩვენც სწორედ ამის გათვალისწინებით, ზემოთ აღწერილი ტექნოლოგიების გამოყენებით შევქმენით სათანადო ვებ-გვერდი შემდეგი ფუნქციონალით: ვებ-გვერდის სტუმარს შეუძლია გაეცნოს სიახლეების ფორუმს, რომელსაც საიტის ადმინისტრატორი აქვეყნებს. საიტის ადმინისტრატორს შეუძლია სიახლეების ფორუმში ახალი ინფორმაციის დამატება, ძველი განცხადებების რედაქტირება ან საჭიროებისამებრ წაშლა. სტუმარს აგრეთვე შეუძლია გამოქვეყნებული სტატიების ნახვა რომლებიც კატეგორიების მიხედვით არიან დაჯგუფებულები. ამჟამად გვაქვს ხუთი კატეგორია: global networks, big data, machine learning, computer vision, software

development. ყველა სტატია, რომელიც კომფერენციის ვებ-გვერდზე აიტვირთება აუცილებლად იქნება მოთავსებული ამ ხუთი კატეგორიიდან ერთ-ერთში. სტუმარს აგრეთვე შეუძლია ნახოს ინფორმაცია კომფერენციის შესახებ და აგრეთვე გაეცნოს საკონტაქტო ინფორმაციას. საკონტაქტო ინფორმაციასთან ერთად დატანებული რუკა, რომელზეც აღნიშნულია ყველა ის ტერიტორია, რომლებსაც კომფერენციასთან რაიმე კავშირი აქვს.

თუ ვებ-გვერდის სტუმარს სურს რომ თავადაც მიიღოს მონაწილეობა კონფერენციაში და გამოაქვეყნოს თავისი სტატია ამისათვის მას სჭირდება რომ დარეგისტრირდეს სამეცნიერო კონფერენციის ვებ-გვერდზე. ამისათვის ის ირჩევს Sign In ფუნქციას რის შედეგადაც გადადის საიტის მომხმარებელთა შესვლის გვერდზე, სადაც აირცევს ველს create an account. ამ ეტაპზე მას მოუწევს რამდენიმე ველის შევსება, კერძოდ: Email, Password, conform password, first name, last name და ბოლო ველში დროშ ბოქსიდან აიეჩიოს ერთ-ერთი ოთხიდან: student, professor, researcher ან other. ამ ველების შევსებისა და დადასტურების შემდეგ, მომხმარებელს უკვე ექნება საკუთარი პროფილი ვებ-გვერდზე და გაუჩნდება შემდგომი ფუნქცია - My Articles. მანდ გადასული მომხმარებელი აღმოჩნდება მის პირად გვერდზე სადაც არის მის შესახებ განთავსებული ინფორმაცია, აქ მას შეუძლია შეცვალოს მისი ვებ-გვერდზე არსებული პროფილის პაროლი და რაც მთავარია ნახოს საკუთარი სტატიები და შექმნას ახალი. ღაც შეეხება ახალი სტატიის შექმნას, საკუთარი პროფილის გვერდიდან ირჩევს ველს create new. ამის შემდეგ მომხმარებელი გადავა სტატიის შექმნის ფორმაზე. მაგ მომენტში ის აირჩევს სტატიის სახელს, მიუთითებს ავტორებს, დროშბოქსიდან აირჩევს 5 არსებული კატეგორიიდან ერთ-ერთს. ამის შემდეგ მას აქვს გამოყოფილი 2 ველი, ერთი იმისთვის რომ განათავსოს ტექსტი და მეორე ველი კომენტარისთვის, რომელიც სტატიასთან ერთად არ გამოქვეყნდება და ის განკუთვნილია მხოლოდ ადმინისტრაციისათვის. ამ ველების შევსების შემდეგ თუ დაასრულა სტატიის რედაქტირება დააჭერს ველს create ამის შემდეგ ეს არსებული სტატია შეინახება მის პირად გვერდზე. აქ მას მის მიერ შექმნილ სტატიებთან მიმართებაში შეუძლია განახორციელოს სამი მოქმედება: details, edit, delete. Details არჩევის შემთხვევაში მომხმარებელი გადავა ახალ გვერდზე სადაც განთავსებული იქნება ინფორმაცია მის მიერ არჩეულ კონკრეტულ სტატიაზე. Delete წაშლის მიმდინარე სტატიას, ხოლო ველი edit გადაგიყვანს ახალ გვერდზე სადაც შეგეძლება შეასწორე ინფორმაცია სტატიის ველებში, ატვირთო სტატიასთან ერთად pdf ან word ფაილები შესაბამისად გაფართოებებით .pdf ან .docx და საბოლოოდ ამავე გვერდიდან გამოაქვეყნოს სტატია და ხარისხის კონტროლის გასავლელად გადააგზავნოს ადმინისტრატორთან.

ვებ-გვერდის მომხმარებლის მეორე კატეგორიაა ადმინისტრატორი, მას რიგითი მომხმარებლის ფუნქციონალის გარდა კიდევ აქვს დამატებითი უფლებები და ფუნქციები. კერძოდ, მხოლოდ მას შეუძლია განათავსოს სიახლეები დაფაზე და მხოლოდ მას შეუძლია მომხმარებელთა მიერ ატვირტულ სტატიებს მიაწიოს სტატუსი approved რაც ნიშნავს რომ აღნიშნული სტატია საჯაროდ გამოქვეყნდება შესაბამისი კატეგორიის ქვეშ. ადმინისტრატორი ერთადერთია ვისაც აქვას წვდომა Admin Panel-თან, სადაც მას აქვს 3

უნიკალური ველი. News- საიდანაც იმართება სიახლეების დაფა, Articles - სადაც განთავსდებიან ის არტიკლები რომლებიც მათმა ავტორებმა ჩადვალეს რომ მზად არიან გამოსაქვეყნებლად და ელიან ადმინისტრატორის ვერდიქტს. აქ მას შეუძლია გაცნოს მომხმარებლის არტიკლს ისეთი ფორმით როგორც მომხმარებელს აქვს საკუთარ გვერდზე, და მაინიჭოს მას ორი ველიდან ერთ-ერთი approved ან rejected. ასევე მას აწვს სპეციალური ველი გამოყოფილი იმისათვის რომ აუხსნას მომხმარებელს მიღებული გადაწყვეტილების შესახებ. Rejected სტატუსის მქონე სტატიები მხოლოდ მისი ავტორის გვერდზე დარჩება, ხოლო accepted სტატუსის მქონე სტატიები გამოქვეყნდება საჯაროდ შესაბამისი კატეგორიის ქვეშ არსებული სახით.

დასკვნა

თუ ვიხელმძღვანელებთ ზემოთ აღწერილი პრინციპებით და პრობლემების გადაჭრის გზებით, ჩვენ მივიღებთ გამართულ მომხმარებლის ინტერფეისს, რომელიც იქნება რეაქტიული, რესპონსივული და კარგი დიზაინის მქონე. თანამედროვე, ჩვენი პროექტის გამოწვევებისათვის საპასუხოდ სათანადოდ შერჩეული ხელსაწყოების საშუალებით შექმნილი ქროს-ბრაუზერული, ქროს-პლატფორმული კლიენტის აპლიკაცია რომელიც ხარვეზების გარეშე შეძლებს ფუნქციონირებს სხვადასხვა სახის ელექტრონულ მოწყობილობებზე.

ლიტერატურა

1. Essential C# 7.0 (6th Edition) (Addison-Wesley Microsoft Technology Series)
2. Microsoft SQL Server 2012 T-SQL Fundamentals (2012), by Itzik Ben-Gan
3. SQL Server 2012 T-SQL Recipes, 3rd Edition (2012), by Jason Brimhall, David Dye, Jonathan Gennick, Andy Roberts, and Wayne Sheffield
4. C# 5.0 in a Nutshell: The Definitive Reference (By: Joseph Albahari, Ben Albahari)
5. introducing razor syntax <https://docs.microsoft.com/en-us/aspnet/web-pages/overview/getting-started/introducing-razor-syntax-c>
6. Entity Framework (EF) Documentation [https://msdn.microsoft.com/en-us/library/ee712907\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/ee712907(v=vs.113).aspx)

დანართი

პროექტი დაწერილია visual studio community 2017-ში. მის გასაშვებად საჭიროა visual studio-ს აღნიშნული ან მასთან თავსებადი ვერსია. პროექტის გასაშვებად საკმარისია visual studio-ში პროექტის გახსნა და IIS სერვერის გაშვება

