

ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი

ჯგუფური პროექტი

დისტანციური სწავლების პორტალი

საბაკალავრო პროგრამა - კომპიუტერული მეცნიერება

პროექტის შემსრულებლები:

სანდრო მირიანაშვილი

დავით ნიკოლეიშვილი

სოფიკო ფხიკიძე

გიორგი პოღოსიანი

ხელმძღვანელი: პროფესორი მანანა ხაჩიძე

თბილისი 2019

ანოტაცია

ელექტრონული სწავლება დაფუძნებულია განათლების სფეროში ინფორმაციული და კომუნიკაციური ტექნოლოგიების გამოყენებაზე. ელექტრონული სწავლება მოიცავს: კომპიუტერის მეშვეობით სწავლებას, დისტანციურ სწავლებას, მობილურ სწავლებას და სხვა მეთოდებს, რომელთა დახმარებითაც ტრადიციული სასწავლო პროცესი იღებს ციფრულ სახეს. მაღალი ინტერაქტიულობის მისაღწევად გამოიყენებს ტექსტურ, აუდიო, ვიდეო მასალებს. ელექტრონულ სწავლებას ახალ საფეხურზე აჰყავს ტრადიციული სწავლების ფორმები.

Electronic portal for distance learning

E-learning is based on the use of information and communication technologies in education field. E-learning includes: computer learning, distance learning, mobile training and other methods that help traditional learning process take digital look. Use text, audio and video materials to achieve high interactivity. Electronic learning forms in the new cycle take the forms of traditional teaching.

**სარჩევი**

|  |  |
| --- | --- |
| ანოტაცია | 2 |
| შესავალი | 4 |
| ვებ აპლიკაცია | 7 |
| ამოცანის დასმა | 8 |
| ამოცანის დაყოფა შრეებად | 9 |
| Front End | 11 |
| Back End | 12 |
| მონაცემთა ბაზა | 14 |
| ამოცანის პრაქტიკული რეალიზება | 15 |
| დასკვნა | 22 |
| ლიტერატურა | 23 |
| სამუშაო გეგმა | 24 |

**შესავალი**

რა არის დისტანციური სწავლება?

ის გახლავთ სწავლების ფორმა, რომელშიც შექმნილია სპეციალურად სტუდენტებისთვის ან სტუდენტთა ჯგუფებისთვის, რომლებიც ტრადიციულ აუდიტორიაში ან კაბინეტში ფიზიკურად არ იმყოფებიან. ეს არის პროცესი, რომელიც ჰქმნის ხელმისაწვდომს სწავლას, როდესაც ინფორმაციის წყარო და მიმღები ერთმანეთისგან დროით, სივრცით, ან ორივე შემთხვევაში დაშორებულები არიან.

დისტანციური სწავლების სათავედ შეგვიძლია 1728 წელი მივიჩნიოთ, როდესაც პირველად გაზეთში **Boston Gazette** განცხადება გაჩნდა, მასწავლებელი ეძებდა მსურველებს, რომლებსაც მასალას კვირაში ერთხელ გაუგზავნიდა. თუმცა დღეს ცნობილი არ არის, იპოვა თუ არა მან სტუდენტები, მაგრამ ის ნამდვილად ცნობილია, რომ ეს მეთოდი აქტიურად მე-19 საუკუნეში, საფოსტო სისტემის დახვეწის შემდეგ ამოქმედდა, მინიმუმ 1840-იან წლებში, როდესაც დიდ ბრიტანეთში ისააკ პიტმანი, ფოსტის მეშვეობით ასწავლიდა სტუდენტებს. ლონდონის უნივერსიტეტი (**University of London**) აცხადებს, რომ სწორედ მან დანერგა პირველად ოფიციალური დისტანციური სწავლება 1858 წელს. მას შემდეგ და მეოცე საუკუნის ბოლომდე ეს სისტემა უფრო და უფრო ვითარდებოდა და იხვეწებოდა, თუმცა ფეხი მხოლოდ კომპიუტერებისა და ინტერნეტის ფართოდ გავრცელების შემდეგ მოიკიდა. 2006 წლის მონაცემებით ყველაზე დიდი სასწავლო დაწესებულებების 96 პროცენტი სტუდენტებს რაიმე სახის ელექტრონულ კურსს სთავაზობს და რომ 2005 წელს 3.2 მილიონმა ამერიკელმა სტუდენტმა ერთი სემესტრი მაინც სწორედ დისტანციური სწავლებით დახურა.

ელექტრონული სწავლება დაფუძნებულია განათლების სფეროში ინფორმაციული და კომუნიკაციური ტექნოლოგიების გამოყენებაზე. ელექტრონული სწავლება მოიცავს: კომპიუტერის მეშვეობით სწავლებას, დისტანციურ სწავლებას, მობილურ სწავლებას და სხვა მეთოდებს, რომელთა დახმარებითაც ტრადიციული სასწავლო პროცესი იღებს ციფრულ სახეს. მაღალი ინტერაქტიულობის მისაღწევად გამოიყენებს ტექსტურ, აუდიო, ვიდეო მასალებს. ელექტრონულ სწავლებას ახალ საფეხურზე აჰყავს ტრადიციული სწავლების ფორმები:

* ლექცია, პრაქტიკულები და სემინარები
* დებატები, სწავლება სოციალური აქტივობის დახმარებით, დისკუსიები და გუნდური სამუშაოები.
* ექსპერიმენტები(ლაბორატორიული სამუშაოები)
* კვლევები და შეფასებები.

დღეს დიდი პოპულარობით სარგებლობს ელექტრონული სწავლება. მისი მომხმარებლების რიცხვი ყოველდღიურად იზრდება. არსებობს უამრავი ელექტრონული სწავლების საიტი, როგორებიცაა: **Coursera**, **Udemy**, **Edx, Udacity** და სხვა უამრავი მათგანი. მათ აქვთ უამრავი დადებითი მხარე: მოქნილი გრაფიკი, ნაკლები ფასი ვიდრე პრესტიჟულ უნივერსიტეტში სწავლა, ცოდნის მიღება ნებისმიერ ადგილას და სწრაფი შედეგი. დისტანციური სწავლება ძალიან გამოადგება არა მარტო თბილისის სახელმწიფო უნივერსიტეტის სტუდენტებს, არამედ ნებისმიერ დაინტერესებულ პირს, რომელიც მზადაა აიმაღლოს ცოდნა, განვითარდეს ამა თუ იმ დისციპლინაში და საგანში. ელექტრონულ სწავლებას აქვს როგორც დადებითი ასევე უარყოფითი მხარეები.

დადებითი:

1. **მოხერხებულობა და ხელმისაწვდომობა** - ელექტრონული კურსების აღება შეგიძლიათ მაშინ, როდესაც ეს მოგესურვებათ, არ არის აუცილებელი სასწავლებლის საგაზაფხულო, საშემოდგომო ან რომელიმე სხვა სემესტრის ლოდინი. ელექტორნული სწავლებისას მუშაობა ნებისმიერი ადგილიდან შეგიძლიათ, თუ თქვენ კომპიუტერი და ინტერნეტი გაქვთ.
2. **მოქნილობა** - არ არსებობს განსაზღვრული დრო ლექციებისთვის, თვითონ წყვეტთ, როდის გსურთ დავალებების შესრულება. თქვენ იცით, როდისთვის რა გაქვთ ჩასაბარებელი, ბევრი კურსი კი გრაფიკის შედგენას თვითონ გთავაზობთ.
3. **არჩევანი** - ბოლო რამდენიმე წლის მანძილზე დისტანციური სწავლების შესაძლებლობების ბუმია, რის შედეგადაც ძალიან ბევრმა აკრედიტირებულმა და კარგი რეპუტაციის მქონე უნივერსიტეტმა დაიწყო ასეთი პროგრამების შედგენა.
4. **ღირებულება** - ფასები, ძირითადად ბევრად ნაკლებია ელექტრონული კურსის არჩევის შემთხვევაში, ვიდრე ეს ტრადიციულ სასწავლებლებშია - თანაც, რა თქმა უნდა, მცირდება კვების, საცხოვრებელი ადგილისა და მგზავრობის ხარჯები.
5. **მართვადი** - სასწავლო გარემოს თვითონ იქმნით, თქვენ არჩევთ ადგილსა და სქემას.
6. **ზოგიერთი მოსწავლისთვის უკეთესია** - მაგალითად ინტროვერტი ადამიანებისთვის, მათთვის ვისაც ენობრივი ბარიერი აქვს, ვისაც მასალის ასათვისებლად მეტი დრო და ძალისხმევა სჭირდება.
7. **სამსახური/კარიერა** - რადგანაც დისტანციური სწავლება კიბერსამყაროშია და თქვენითვე რეგულირდება, არ არსებობს აუცილებლობა შეწყვიტოთ მუშაობა, ან აიღოთ აკადემიური და ამით გაიფუჭოთ საქმე ან სამსახურში ან სასწავლებელში.

უარყოფითი:

1. **ლექტორთან პირისპირ შეხვედრა** - თუ თქვენ ყოველთვის მასწავლებელთან, ლექტორთან ურთიერთობა გირჩევნიათ და ყურადღება გჭირდებათ, სავარაუდოდ სწავლების ეს სისტემა თქვენთვის არ არის.
2. **აღქმა/რეპუტაცია** - მიუხედავად იმისა, რომ ნელ-ნელა მეტი და მეტი ცნობილი უნივერსიტეტი ჰქმნის ელექტრონულ პროგრამას, მაინც არსებობს გარკვეული სტერეოტიპი ამ ტიპის სწავლების მიმართ.
3. **მოითხოვს ახალ უნარებსა და ტექნოლოგიებს** - თუკი თქვენ არ გეხერხებათ კომპიუტერთან და ინტერნეტთან ურთიერთობა ან საერთოდ არ გიყვართ თანამედროვე ტექნოლოგიები, დისტანციური სწავლება თქვენთვის არ არის.
4. **არანაირი სოციალური ინტერაქცია** - მიუხედავად იმისა, რომ თანაკურსელებთან ელექტრონული მეილის, ჩათებისა და ფორუმების მეშვეობით შეგიძლიათ და მოგიწევთ კიდეც კონტაქტი, მაგრამ ეს პირად ურთიერთობას მაინც ვერ შეცვლის.
5. **დროის გაყვანა** - თუ თქვენ ადამიანების იმ ტიპს მიეკუთვნებით, რომელსაც ყოველთვის სჭირდება ბიძგი და გარემო, რომ სამუშაო დაამთავროს, ალბათ ძალიან გაგიჭირდებათ დისტანციური სწავლება.
6. **ნაკლები მხარდაჭერა** - სტუდენტებისგან მოითხოვება სხვადასხვა რესურსის მოძიება დავალებების შესასრულებლად, რაც ზოგისთვის სასიამოვნო შესაძლებლობაა, ზოგისთვის კი დამთრგუნველი და შემაშფოთებელი.
7. **სტუდენტური ცხოვრება** - ბევრისთვის სტუდენტობა მნიშვნელოვან და სახალისო პერიოდთან ასოცირდება, მოიცავს ახალი მეგობრებისა და ნაცნობების შეძენას, მათთან ერთად გართობას და ა.შ დისტანციური სწავლება კი სტუდენტურ ცხოვრებას საერთოდ არ მოიცავს.

**რა არის ვებ აპლიკაცია?**

დღესდღეობით დიდი პოპულარობით სარგებლობს web აპლიკაციები, რასაც ვერ ვიტყვით Desktop აპლიკაციებზე. იგი გამოწვეული შემდეგი მიზეზების გამო: პორტაბელორობა (Portable), ერთხელ დაწერილი web აპლიკაცია გაეშვება ნებისმიერ მოწყობილეობაზე და სისტემაზე, არ არის საჭირო მათი განახლება, რადგან Server-ზე იქნება ატვირთული უახლესი ვერსია, ადვილია შედლომების (Bug) გასწორება, ადვილია პროდუქტის მხარდაჭერა (Support), უპრობლემოდ იმუშავებს მობილურ მოწყობილეობაზე (Mobile Device). დღეს ძალიან ბევრ ორგანიზაციის აქვს საკუთარი web აპლიკაცია, რომელსაც იყენებს სერვისებისთვის, ბიზნესისთვის, ფინანსების მართვისთვის და უამრავი სხვა მიზნებისთვის.

Web აპლიკაცია არის პროგრამა, რომელიც ინახება სერვერზე და მასთან კონტაქტს ვამყარებთ ბრაუზერის (Google Chrome, Firefox) მეშვეობით. ნებისმიერ პროგრამას შეგვიძლია დავუკავშირდეთ ბრაუზერით. ჩვენს გარშემო უამრავი ვებ აპლიკაციაა, რომელთაც ყოველდღიურად ვიყენებთ მაგ: facebook.com, google.com. ასეთ პროგრამებს აქვთ ბევრი დადებითი მხარე.

1. ვწერთ ერთხელ და ვიყენეთს ყველგან (**develop one and run anywhere**) - web აპლიკაციის არქიტექტურა აძლევს დეველოპერს საშუალებას შექმნას აპლიკაცია ერთხელ და მოხმარებელმა/კლიენტმა გამოიყენოს ნებისმიერ მოწყობილეობაზე, რომელიც დაკავშირებულია ინტერნეთან (მაგალითად Smart phone, PC და ლეპტოპი). იგი ყველა მოწყობილობის მოდელზე, ტიპზე, ოპერაციულ სისტემაზე მუშაობს ერთნაირად.
2. ადვილად განახლებადი (**Easy to upgrade**) - Web application არის ძალიან ადვილად განახლებადი. მისი განახლებისთვის/გაუმჯობესებისთვის საჭიროა დეველოპერმა შეცვალოს კოდი სერვერზე და ავტომატურად ყველა მოხმარებელს ექნება საშუალება ნახოს ახალი პროდუქტი.
3. რამდენად ეფექტურად მუშაობს აპლიკაცია (**Performance**) - Web აპლიკაცია არის ძალიან კარგი საშუალება, რომ შევქმნათ დიდი ზომის პროგრამები (Enterprise Level Software).
4. ღია კოდის პროგრამები (**Open source Software**) - არსებობს უამრავი ხელსაწყოები (tools), ბიბლიოთეკები (Libraries and Frameworks), რომლებიც გვეხმარება web აპლიკაციების სწრაფად შექმნასა და ბიუჯეტის დაზოგვაში. მაგალითად github-ის პროექტები: Bootstrap(<https://github.com/twbs/bootstrap>), Angular(<https://github.com/angular/angular>) და სხვა უამრავი მათგანი.

**ამოცანის დასმა**

ელექტრონული სწავლების პორტალი არის ინოვაციური პლატფორმა, რომელიც სტუდენტს საშუალებას აძლევს მიიღოს ცოდნა ინტერნეტის საშუალებით. მათ შეუძლიათ აირჩიონ კურსები. თითოეული მათგანის ხანგრძლივობაა 15 კვირა, რომელიც შეიცავს სასწავლო მასალებს და დავალებას. იმისთვის, რომ სტუდენტმა შეძლოს დავალების შესრულება, საჭიროა კარგად გაარჩიოს სასწავლო მასალა და ამის შემდეგ გადავიდეს დავალების შესრულებაზე. სტუდენტი, როდესაც დაასრულებს დავალების შესრულებას, ამ ყველაფერს ატვირთავს ჩვენს პორტალზე. მის მიერ ატვირთულ ფაილს ნახავს კურსის ლექტორი და შეაფასებს გარკვეული ქულით. კურსის მონაწილეებმა ასევე უნდა იცოდნენ, რომ თითოეულ დავალებას აქვს თავისი ვადა (**deadline**), რომლის დარღვევის შემთხვევაში მიიღებენ ნულ ქულას (დავალების ატვირთვა აღარ შეეძლებათ). პორტალზე ლექტორის მოვალეობა იქნება შემდეგი: თითოეული კვირის მასალა მიაწოდოს თავის მსმენელებს, შექმნას გარკვეული ტიპის დავალებები და შეაფასოს ისინი.

დღეს საქართველოში არც თუ ისე პოპულარულია ელექტრონული სწავლება. ხალხს ურჩევნია ცოდნის მისაღებად ტრადიციულ წესს მიმართონ. ჩვენმა ჯგუფმა იცის, რომ ბევრი ადამიანი იყენებს ინტერნეტს, ამიტომ გადავწყვიტეთ დისტანციური სწავლების ვებ გვერდი შევქმნათ. მასთან დაკავშირებას ნებისმიერი ადამიანი შეძლებს, ვისი მოწყობილობაც ინტერნეტს უკავშირდება, არა აქვს მნიშვნელობა მობილური ექნება თუ პერსონალური კომპიუტერი. ჩვენი დასახული ამოცანის უკეთ გადასაწყვეტად საჭიროა ის დავყოთ შემდეგ ნაწილებად:

1. საიტის ინტერფეისი (**Frontend**)
2. აპლიკაციის ლოგიკა (**Backend**)
3. მონაცემთა ბაზა (**Database**)

თითოეულ ნაწილს დეტალურად განვიხილავთ შემდეგ პუნქტებში.

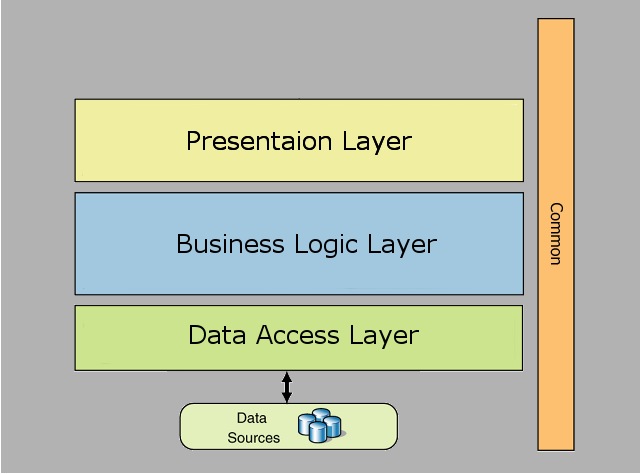
ელექტრონული სწავლების ჯგუფური პროექტის გუნდი შედგება 4 სტუდენტისგან. თითოელი მათგანს გადანაწილებული აქვს თავისი მოვალეობები, საქმე, შემდეგი პრინციპით:

1. **სანდრო მირიანაშვილი** - backend დეველოპერი, ძირითადად იმუშავებს პორტალის ბიზნეს ლოგიკაზე და გამოიყენებს asp.net mvc ფრეიმვორკს.
2. **დავით ნიკოლეიშვილი** - მონაცემთა ბაზების დეველოპერი. ის იმუშავებს მონაცემთა ბაზების შექმნაზე, მოდერნიზაციაზე და განვითარებაზე. დავითი გამოიყენებს Microsoft Sql Server.
3. **გიორგი პოღოსიანი** - Frontend დეველოპერი. მისი ძირითადი მოვალეობაა შექმნას საიტის მომხმარებლის ვიზუალური მხარე html/css გამოყენებით.
4. **სოფიკო ფხიკიძე** - Frontend დეველოპერი, შექმნის ლექტორის და სტუდენტის პირად გვერდს html/css და javascript-ის გამოყენებით.

**ამოცანის დაყოფა შრეებად**

N შრიანი არქიტექტურა ჰყოფს ჩვენს აპლიკაციას N ნაწილად, სადაც N-მა შეიძლება მიიღოს 1-ზე მეტი მნიშვნელობა. ყველაზე ხშირად გამოიყენებენ 3 შრიან არქიტექტურას, შემდეგი კომპონენტებით: **Presentation Tier**, **Application Tier** და **Database Tier**.

* **Presentation Tier** - სამომხმარებლო ინტერფეისი. მისი მთავარი მოვალეობაა გარდაქმნას დავალებები და შედები ისე, რომ ეს ყველაფერი მომხმარებლისთვის გასაგები იყოს.
* **Application Tier** - ეს შრე იღებს აპლიკაციიდან ბრძანებებს, იღებს ლოგიკურ გადაწყვეტილებებს, ასრულებს გამოთვლებს. ის არის ერთგვარი შუამავალი ამ ორ შრეს შორის.
* **Database Tier** - ინფორმაცია, რომელსაც ვინახავთ ან ვიღებთ მონაცემთა ბაზიდან ან ფაილური სისტემიდან.



ჩვენი Web აპლიკაციის შესაქმნელად გამოვიყენოთ 3-შრიანი არქიტექტურა (3 tier architecture) და მას დავყოფთ ზუსტად ისე, როგორც სურათზეა ნაჩვენები.

**Presentation Tier** - არის საიტის სამომხმარებლო ინტერფეისი. მის გასაკეთებლად გამოვიყენებთ თანამდედროვე HTML5, CSS3 და Javascript ტექნოლოგიას. ეს შრე შეგვიძლია გავყოთ 4 ლოგიკურ მოდულად: აპლიკაციის ძირითადი ინტერფეისი (მთავარი გვერდი, ავტორიზაციის გვერდი), ადმინისტრატორის სამართავი პანელი, სტუდენტის და ლექტორის პირადი გვერდი.

**Application Tier** - ამოცანის გადასაწყვეტად გამოვიყენებთ Asp.Net Mvc ტექნოლოგიას. იგი არის web აპლიკაციების ფრეიმვორკი, რომელიც დამზადებულია Microsoft-ის მიერ. Asp.Net გამოიყენებს პროგრამული არქიტექტურის საკმაოდ ცნობილ Mvc პატერნს.

**Database Tier** - Web აპლიკაცია მონაცემებთან სამუშაოდ გამოვიყენოთ Microsoft SQL Server. იგი არის რელაციური მონაცემთა ბაზა.

**Front End**

ყოველდღე მომხმარებელი აქტიურად იყენებს ინტერნეტს, ეცნობა სიახლეებს, სტუმრობს მრავალ ვებგვერდს. ალბათ, საკამათო არ არის, რომ თითოეული მომხმარებლისთვის მნიშვნელოანია კარგი და გამართული ვებგვერდი, როგორც არქიტექტურის, ასევე დიზაინის მხრივ. ეს მათთვის ერთგავრი კომფორტიცაა.

ვებ აპლიკაციის ერთ-ერთი ძირითადი ნაწილია მისი ვიზუალური მხარე. მნიშვნელოვანია, რომ პროდუქტს გააჩნდეს სტრუქტურულად კარგად აგებული დიზაინი, რადგან ზუსტად ინტერფეისის საშუალებით ხდება აპლიკაციასთან ურთიერთქმედება. ის გამოცდილება, რომელსაც იღებს მომხმარებელი აპლიკაციის სარგებლობისას, ეწოდება UX (User Experience) დიზაინი. UX დიზაინის ჩამოყალიბება მოიცავს როგორც ვიზუალური ნაწილის (UI) აგებას, ასევე ინტერფეისის სტრუქტურას. სწორად აგებული დიზაინი გაუმარტივებს მომხმარებელს აპლიკაციით სარგებლობას, რაც აისახება მის გამოცდილებაზე.

ვებ აპლიკაციის დიზაინის აგება იწყება გარკვეული სახის გამოკვლევებით და ანალიზით. პირველ რიგში უნდა გადაწყდეს ზოგადი საკითხები: აპლიკაციის დანიშნულება, მომხმარებლების ტიპების განსაზღვრა, სისტემური მოდელების შექმნა და სტრუქტურის აგება. შემდეგი ეტაპი მოიცავს ტექნოლოგიების შერჩევას, რითაც შესაძლებელი იქნება განხილული საკითხების პროექტის სახით ჩამოყალიბება.

ნებისმიერი ვებ-გვერდის შესაქმნელად საჭიროა გარკვეულ ინსტრუმენტთა ნაკრები, რომელთა ერთობლიობა მოგვცემს სასურველ შედეგს. Front-End, ანუ მომხმარებლის მხარე, მოიცავს ტექნოლოგიებს სამომხმარებლო ინტერფეისის ასაგებად. გამოვყოთ ძირითადი ტექნოლოგიები, რომლებიც გამოიყენება ჩვენს პროექტში:

* **HTML 5 (Hypertext Markup Language)** - მარკირების ენა, რომლის დანიშნულებაა შექმნას ვებ-გვერდის ძირითადი სტრუქტურა. შედგება სპეციალური ტეგებისგან, რომლებსაც კითხულობს ბრაუზერი ვებ-გვერდზე ელემენტების დასახატად. ჩვენს აპლიკაციაში გამოიყენება HTML-ის ბოლო ვერსია, რომლის გამოყენებით ვებ-საიტის ინდექსირება საძიებო სისტემებში ხდება გაცილებით სწრაფად
* **CSS 3 (Cascading Style Sheets)** - კასკადური სტილების ენა, რომლის დანიშნულებაა HTML ტეგებისთვის სტილების გაწერა, ანუ მათი ვიზუალის შეცვლა (პოზიციონირება, ზომა, ფერი და ა.შ.). CSS-ის ბოლო ვერსია გვაძლევს საშუალებას გამოვიყენოთ ანიმაციები და ისეთი გრაფიკული ეფექტები, როგორებიცაა, მაგალითად, ჩრდილები და გრადიენტი
* **JavaScript (ECMAScript)** - პროგრამირების ერთ-ერთი ფართოდ გავრცელებული ენა, რომლის პოპულარობა ბოლო დროს მნიშვნელოვნად გაიზარდა. ის გვაძლევს საშუალებას შევქმნათ ვებ-გვერდზე დინამიურად განახლებადი და ინტერაქტიული კონტენტი. არსებობს Javascript-ის ბევრი სტანდარტი, რომლებსაც ოფიციალურად ეწოდებათ ECMAScript (შემოკლებულად - ES). ჩვენი აპლიკაცია დაეყრდნობა ES6 სტანდარტს და გამოიყენებს ზოგიერთ უახლეს ფუნქციონალს. დამატებითად გამოვიყენებთ JQuery ბიბლიოთეკას, რომელიც მოგვცემს საშუალებას ვწეროთ უფრო მოკლე და მარტივი კოდი

**Backend**

რა არის Back end?

ეს არის კოდი, რომელიც აკავშირებს ვებ საიტს მონაცემთა ბაზებთან, მართავს მომხმარებელთა კავშირებს, აძლევს ძალას თავად ვებ აპლიკაციის მუშაობას. სტუნტების ელექტრონული სასწავლო პორტალისთვის გამოვიყენებთ asp.net ფრეიმვორკს.

**.Net Framework**

პროგრამული უზრუნველყოფის პლატფორმა, რომელიც შექმნა კომპანია მაიკროსოფტმა. მასზე შექმნილი აპლიკაციები მუშაობს windows ოპერაციულ სისტემაზე. .Net Framework-ის პირველი ვერსია ბაზარზე გამოჩნდა 2002 წელს. მისი გამოყენებით შეგვიძლია შევქმნათ ვებ და დესკტოპ აპლიკაციები. ამ ბიბლიოთეკას აქვს C# და Visual Basic მხარდაჭერა.

**Asp.Net MVC**

Web აპლიკაციების შესაქმნელი ფრეიმვორკი. იგი დაამზადა კომპანია Microsoft-მა. ის გამოიყენებს ცნობილ MVC არქიტექტურულ შაბლონს. Asp.net Mvc გარემო ისეა მოწყობილი, რომ დეველოპერს შეუძლია მარტივად დაწეროს Unit Test-ები თავისი კოდისთვის.

**Design Patterns**

რა არის დიზაინ პატერები (Design patterns)? ეს არის პროგრამული დიზაინის ტიპიური გადაწყვეტილება ხშირად განმეორებად პრობლემებში. პატერნები არ არის დასრულებული დიზაინი, რომელსაც პირდაპირ კოდში გადავიტანთ. ის არის ერთგვარი თარგი (template)

თუ როგორ გადავწყიტოთ ჩვენ მიერ დასმული ამოცანა სხვადასხვა სიტუაციაში. როგორ შეიქმნა ისინი? ხშირად რაიმე სახის პრობლემის გადაჭრის გზა სხვადასხვა ამოცანაში მეორდება. ამ დროს ვიღაც მას მოუფიქრებს სახელს და აღწერს. ამის შემდეგ მივიღებთ პროგრამის დიზაინის თარგს (pattern). ისინი ეხმარება დეველოპერს გადაწყვიტოს პრობლემა მარტივად, დაზოგოს დრო და მას არ მოუწიოს ყოველ ჯერზე ხელახლა გამოიგონოს ბორბალი (re-inventing wheel), რომელიც უკვე სხვამ გამოიგონა.

**MVC Design Pattern**

MVC არის ყველაზე ცნობილი არქიტექტურული პატერნი კომპიუტერულ მეცნიერებაში. იგი შეიქმნა 1979 წელს შემდეგი სახელით Thing-Model-View-Editor. მალე ის გამარატივდა და შეეცვალა სახელი და დაერქვა Model-View-Controller. MVC პატერნი დეველოპერს აძლევს საშუალებას აპლიკაცია გაყოს სამ ნაწილად: Model, View და Controller.

* **Model** - არის პროგრამის ნაწილი, სადაც უნდა გადავწყვიტოთ აპლიკაციის ბიზნეს ლოგიკა. ხშირად მოდელის ობიექტები მოგვაქვს მონაცემთა ბაზიდან. მაგალითად: გვინდა წავიკითხოთ პროდუქტის ობიექტი მონაცმეთა ბაზიდან, განვანახლოთ ის რაიმე პარამეტრით (მაგალითად: ფასით, ოდენობით) და ჩავწეროთ ეს მონაცემები ბაზაში.
* **View** - არის კომპონენტები, რომლებიც ჩნდება აპლიკაციის user interface-ზე. UI იქმნება model-ის მინაცემებით. მაგალითად: ვთქვათ აპლიკაციაში ჩავტვირთეთ პროდუქტის რედაქტირების გვერდი, სადაც გამოჩნდა ველები მანაცემებით შევსებული. ეს მონაცემები (Data) მოდის Model-იდან.
* **Controller** - არის კომპონენტების ერთობლიობა, რომელიც მოქმედებს მომხმარებლის მიერ განხორციელებულ ქმედებაზე და უბრუნებს მისთვის სასურველ შედეგს ეკრანზე. Asp.Net MVC აპლიკაციაში, View არის მხოლოდ ინფორმაციის გამოტანისთვის/ჩვენებისთვის და Controller იჭერს მოხმარებელის მოქმედებას და რეაგირებს მასზე. მაგალითად: კონტროლერი იჭერს (handle) ბრძანების მნიშვნელობას და აწვდის მას Model-ს, რომელიც მოქმედებს მონაცემთა ბაზასთან. მაგ: პროდუქტთა ცხრილიდან ამოვიღოთ ისეთო ობიექტი, რომლის Id არის 100-ის ტოლი.

**მონაცემთა ბაზა**

**მონაცემთა ბაზა** — ელექტრონული ინფორმაციის მართვის სისტემა. მისი მთავარი დანიშნულებაა ინფორმაციის დაცულად შენახვა და მომხმარებლისთვის სათანადო ფორმით მიწოდება. სისტემა შედგება ორი ნაწილისაგან: მონაცემთა ბაზის სამართავი პროგრამისა (**DBMS** - **Database management system**) და თვითონ ინფორმაციისგან. სამართავი პროგრამის დანიშნულებაა ინფორმაციის შენახვა, ორგანიზება, მართვა და გადმოცემა.

დისტანციური სწავლების ერთ-ერთი უმნიშვნელოვანესი უპირატესობა არის ის, რომ მას

შეუძლია უზრუნველყოს ასინქრონული განათლების უზრუნველყოფა ვირტუალურ

გარემოში. ამ ყველაფრისთვის საჭიროა შესაბამისი მოქნილი მონაცემთა ბაზის შექმნა, რომლის სტრუქტურა უნდა იყოს დეტალური შესწავლის ობიექტი, რომელიც მრავალჯერ უნდა გადამოწმდეს შექმნის პერიოდში. ცხრილები და ამ ცხრილების თვისებები უნდა

განიხილებოდეს ნებისმიერ მოდულის სტრუქტურაში ანალიზისა და დიზაინის

პროცესში. ზოგიერთი პროცესის გამო, როგორიცაა ინფორმაციის ფილტრაცია ან

ინფორმაციის ხელმისაწვდომობა, მონაცემთა ბაზის მოდელს უნდა ჰქონდეს მოქნილი

სტრუქტურა.

სასწავლო ტექნოლოგიების შექმნისას, როგორც წესი, იყენებენ

რელაციურ მონაცემთა ბაზებს, როგორებიცაა MS SQL Server, MySql, ORACLE, SQLite და

ა.შ. ჩვენს შემთხვევაში გამოვიყენებთ MS SQL Server-ს.

**SQL Server** რელაციური მონაცემთა ბაზაა, იგი გვაძლევს მონაცემთა შენახვის სანდო სტრუქტურას, რომლის გამოყენებაც მარტივად შეიძლება აპლიკაციებში. თუ თქვენ იწყებთ Express edition - ით, მომავალში შეიძლება გაგიჩნდეთ სურვილი გადახვიდეთ უფრო მძლავრ და გაფართოებულ მონაცემთა ბაზების სისტემაზე, რაც მოიცავს სტრუქტურირებულ სივრცეს, სწრაფ ძებნასა და ინფორმაციის გამოტანას, ტრანზაქციულ წერა - კითხვას, რეპორტების გენერირებას [data analysis](https://social.technet.microsoft.com/wiki/contents/articles/1175.sql-server-analysis-services-ssas.aspx) და დამატებითი საინფორმაციო ტიპების მხარდაჭერას.

აპლიკაციიდან მონაცემებთან მუშაობისას კი ვიყენებთ Entity Framework-ს.

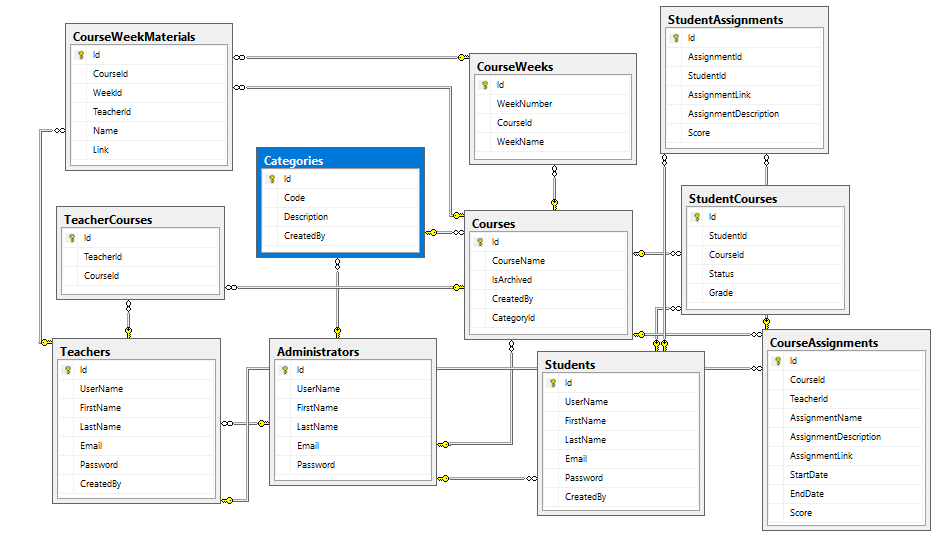
Entity Framework წარმოადგენს .NET ფრეიმვორკზე დამფუძნებულ ობიექტზე ორიენტირებულ ტექნოლოგიას მონაცემებთან სამუშაოდ. Entity Framework გვაძლვს საშუალებას ადვილად დავაკონექტდეთ და ვიმუშაოთ მონაცემთა ბაზებთან, მაგალითად მონაცემების წაშლა, დამატება, ცვლილებების შეტანა, დასელექტება. ასევე ცხრილების შექმნა მონაცემთა ბაზაში, და მოკლედ ყველანაირი საჭირო მანიპულაციის გაკეთება.

**ამოცანის პრაქტიკული რეალიზება**

როგორც წინა თავებში აღვნიშნეთ, პროექტზე მუშაობდა 4 სტუდენტი. ორი სემესტრის განმავლობაში ჩვენ შევძელით შეგვექმნა მონაცემთა ბაზის კომპლექსური არქიტექტურა, მომხმარებლის ინტერფეისი და დემო ვებ აპლიკაცია.

მონაცემთა ბაზის ასაგებად გამოვიყენეთ რელაციური Microsoft Sql Server. მისი არქიტექტურა ისე არის აგებული, რომ შესაძლებელია მისი გაფართოება საკმაოდ ადვილად, რაც შემდგომში დაეხმარებათ ახალ გუნდებს, ჯგუფებს, რომ შეძლონ დაამატონ ამ პროგრამულ უზრუნველყოფას გარკვეული თვისებები, ცხრილები და ახალი მოთხოვნები, რომელებიც გაჩნდება ჩვენი დამკვეთის ან პროექტის განვითარებიდან გამომდინარე.

მონაცემთა ბაზა შემდეგნაირადაა დაპროექტებული:

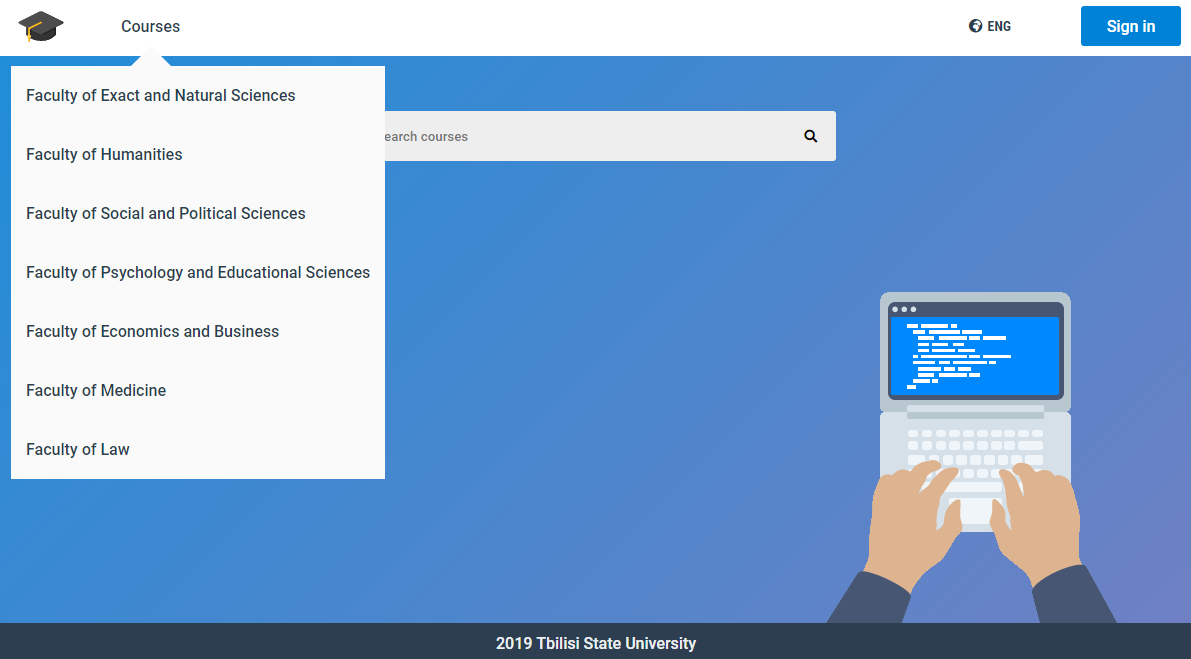


როგორც შემდეგი დიაგრამიდან ჩანს, ჩვენ ვიყენეთ თითოეული ცხრილის დასაკავშირებლას შემდეგ კავშირების ტიპებს: 1..N, 1..1 და N..M. ცხრილების სახელები: კურსები, სტუდენტები, ადმინისტრატორი, კურსის კვირის მასალა, ლექტორები, ლექტორის კურსები, სტუდენტების კურსები, კურსის დავალებები.

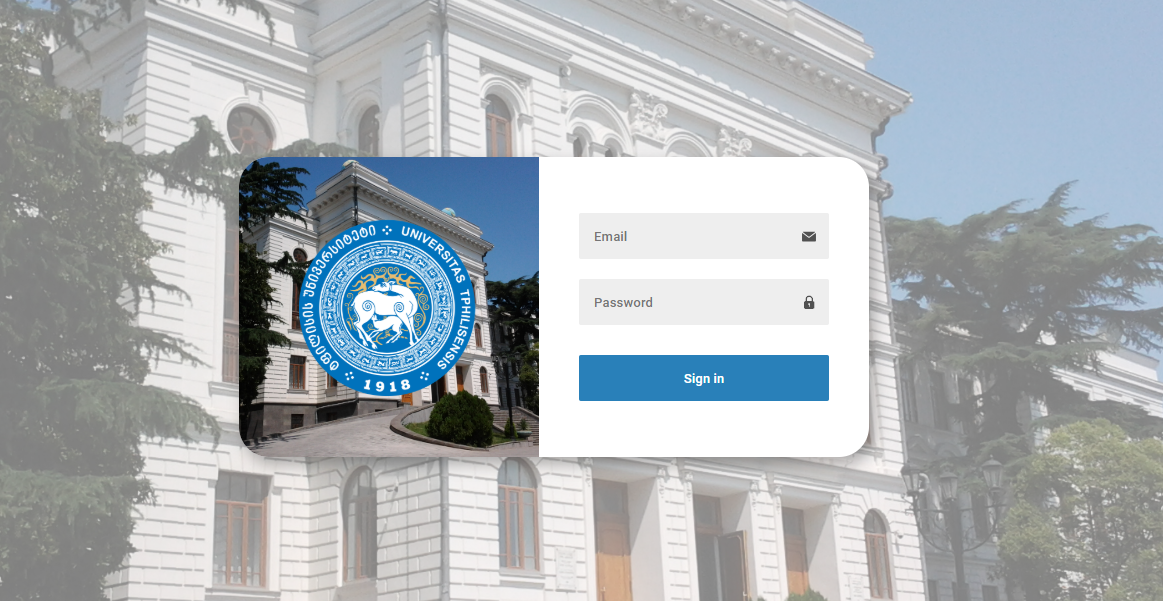
მომხმარებლის ინტერფეისი

ელექტრონული სწავლების აპლიკაცია ხელმისწავდომი რომ იყოს მოხმარებელებითსვის საჭიროა მას ჰქონდეს მარტივი ინტერფეისი. მის შესაქმნდელად გამოვიყენეთ ახალი Html5 და css 3 ტექნოლოგია. Html კეთხდება ვებ აპლიკაციის ჩონჩხი. იმისთვის რომ ის გავალამაზოთ, დავადოთ გარკვეული სტილები, საჭიროა გამოვიყენოთ css 3 ტექნოლოგია.

რამდენიმე სურათი რომელზეც ჩანს ელექტრონული სწავლების ინტერფეისი.



მთავარი გვერდი იგი შედგება შემდეგი კომპონენტებისგან: ლოგო, კურსების ჩამოსაშლელი მენი, უცხო ენის არჩევის ღილაკი და სისტემაში შესვლის ღილაკი.



სისტემაში შესვლის გვერდი. ის შედგება ელ. ფოსტის, პაროლის ველებისგან და შესვლის ღილაკისგან.

ელექტრონული სწავლების პორტალის ასაგებად ვიყენებთ Asp.Net ტექნოლოგიას. აპლიკაცია შეიცავს როგორც მომხმარებლის გვერდებს, ასევე ადმინ. პანელებს შემდეგი ტიპის მომხმარებლებისთვის: ლექტორი, სტუდენტი და ადმინისტრატორი.

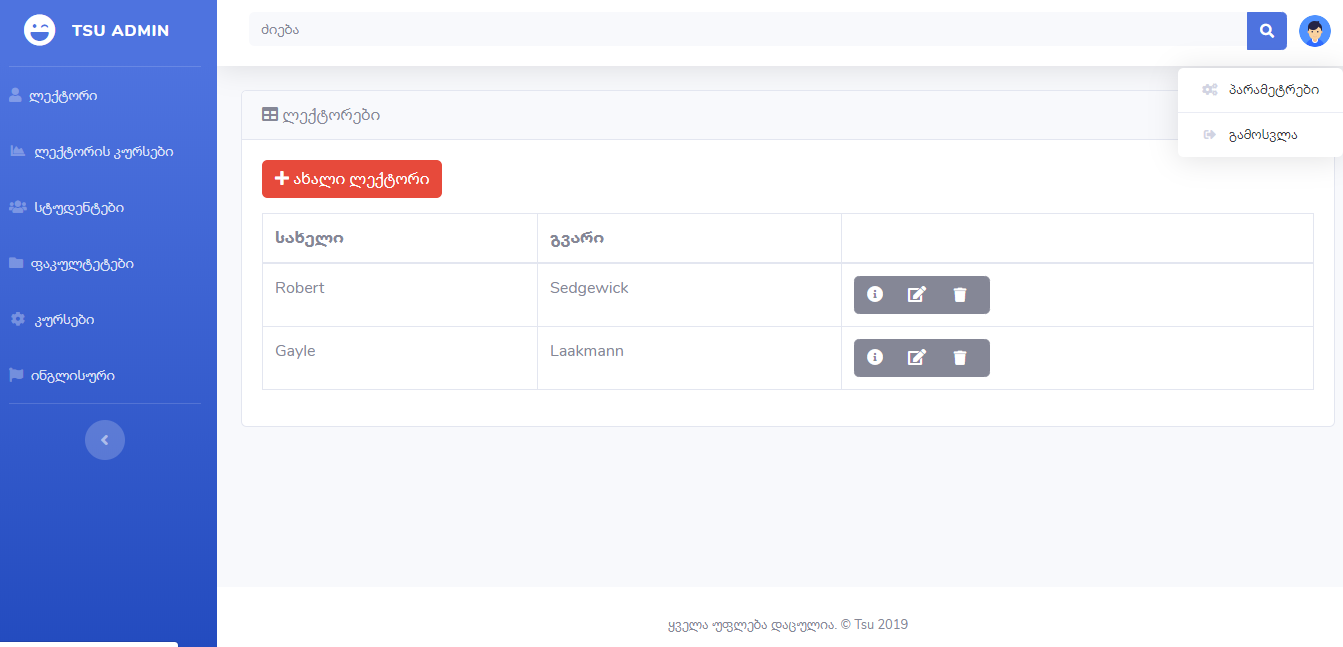
ისინი ავტორიზაციას გადიან სისტემაში შესვლის გვერდიდან (მის შესახებ ვისაუბრეთ ზემოთ სურათში).

სამართავი პანელების ვიზუალური მხარისთვის გამოვიყენეთ კომპანია **Twitter**-ის შექმნილი ბიბლიოთეკა **Bootstrap**. ის დეველოპერს ეხმარება მარტივად შექმნას ლამაზი ვიზუალის მქონე გვერდები თავისი აპლიკაციისთვის.

ელექტრონული სწავლების ვებ აპლიკაციისთვის გამოვიყენეთ ღია კოდის წვდომის (**Open Source**) ინტერფეისის დიზაინის შაბლონი. ის დაგვეხმარა შეგვექმნა სამართავი პანელი ბევრად უფრო ინტერაქტიული, მოქნილი მომხმარებლებისთვის. ის ატვირთულია github-ზე, შემდეგ მისამართზე - <https://github.com/BlackrockDigital/startbootstrap-sb-admin-2>

სხვადასხვა ეკრანის სურათი პროექტის უკეთ სადემონსტრაციოდ.

საიტის ადმინისტრატორის გვერდი



ინტერფეისი ორი ნაწილისგან შედგება მენიუ და მთავარი ნაწილი.

მენიუ

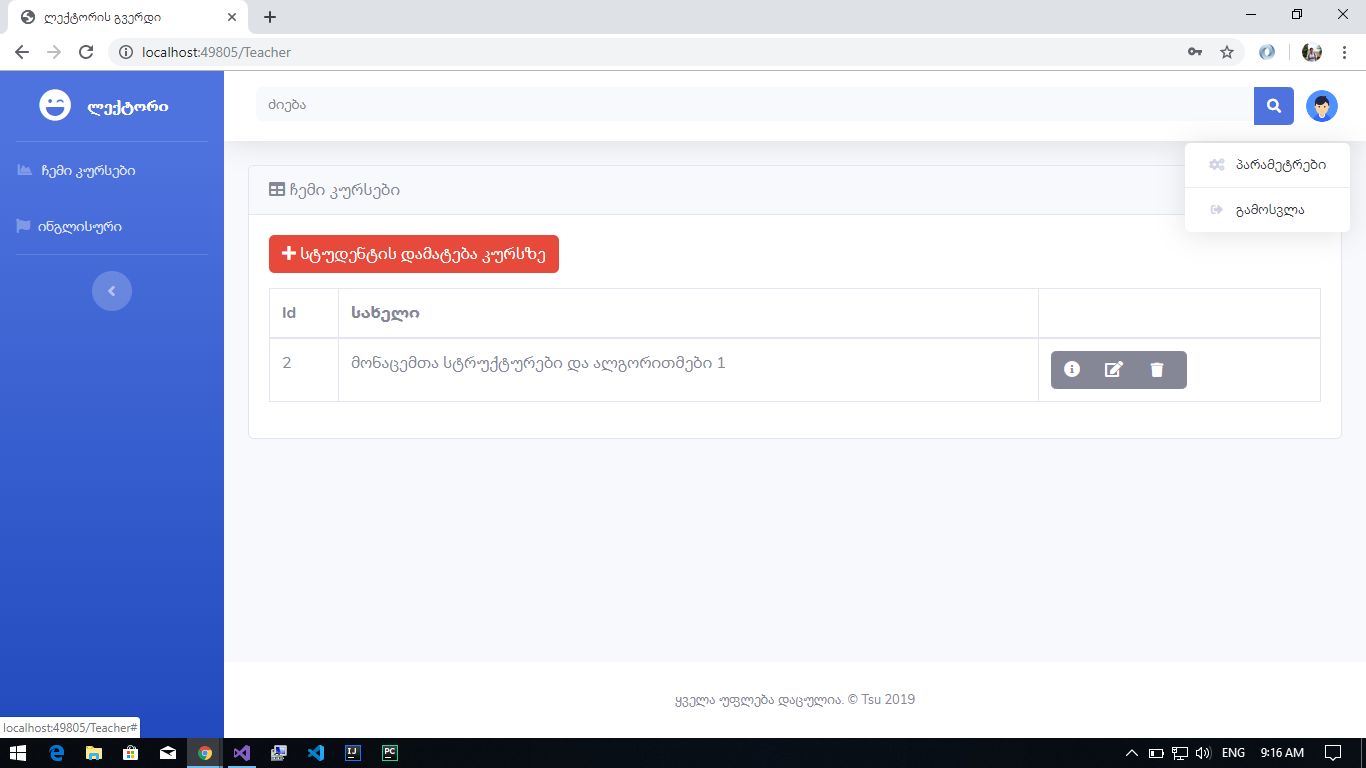
1. ლექტორი
2. ლექტორის კურსები
3. სტუდენტები
4. ფაკულტეტები
5. კურსები
6. ენის ცვლილება - აპლიკაცია არის ორ ენოვანი: ინგლისური და ქართული

მთავარი ნაწილი - რომელ ლინკზეც გადავალთ მენიუდან, იმ ლინკის შესაბამისი ინფორმაცია ჩაიტვირთება ამ ნაწილში.

მომხმარებლის გამოსახულების ღილაკზე დაჭერისას ჩამოიშლება შემდეგი რამ: პარამეტრები და გამოსვლა.

1. პარამეტრები - მომხარებელს შეუძლია შეცვალოს თავისი ინფორმაცია: პაროლი, სახელი და ელ.ფოსტა.
2. გამოსვლა - მომხმარებლის გამოსვლა სისტემიდან.

ლექტორის პირადი გვერდი



ინტერფეისი ორი ნაწილისგან შედგება: მენიუ და მთავარი ნაწილი.

მენიუ

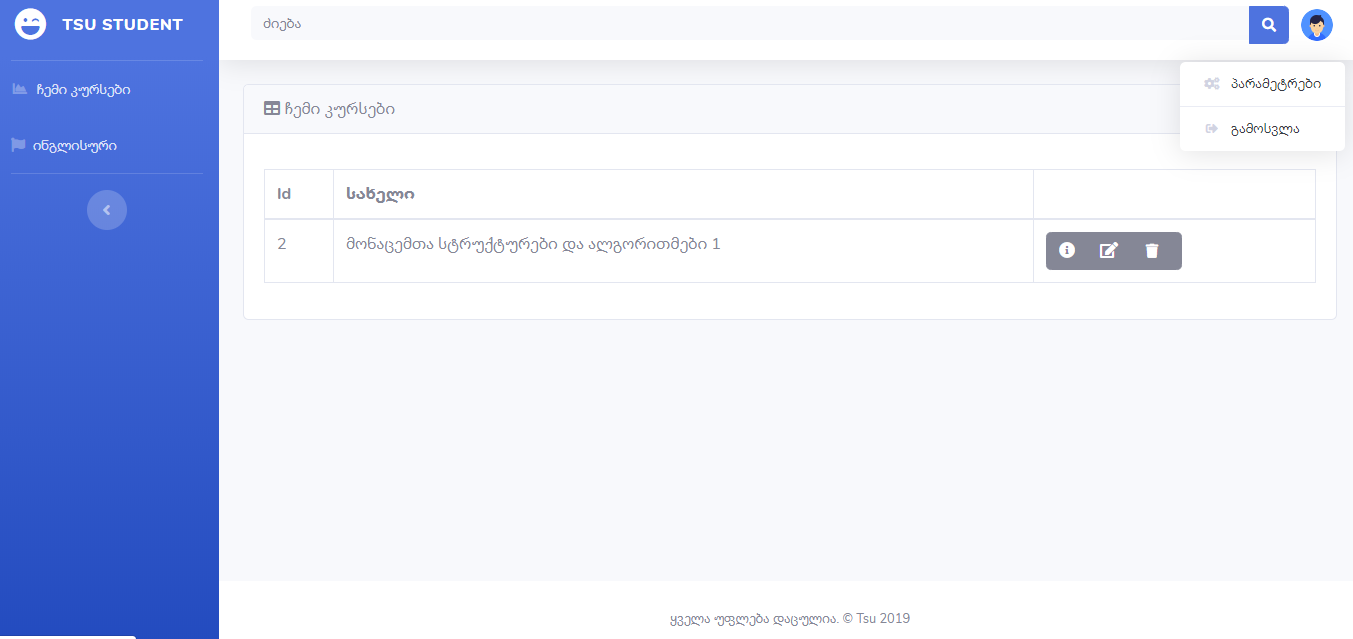
1. კურსები - კურსები, რომელსაც ლექტორი ასწავლის
2. ენის ცვლილება - აპლიკაცია არის ორ ენოვანი: ინგლისური და ქართული

მთავარი ნაწილი - რომელ ლინკზეც გადავალთ მენიუდან, იმ ლინკის შესაბამისი ინფორმაცია ჩაიტვირთება ამ ნაწილში.

მომხმარებლის გამოსახულების ღილაკზე დაჭერისას ჩამოიშლება შემდეგი რამ: პარამეტრები და გამოსვლა.

1. პარამეტრები - მომხარებელს შეუძლია შეცვალოს თავისი ინფორმაცია: პაროლი, სახელი და ელ.ფოსტა.
2. გამოსვლა - მომხმარებლის გამოსვლა სისტემიდან.

სტუდენტის გვერდი



ინტერფეისი ორი ნაწილისგან შედგება: მენიუ და მთავარი ნაწილი.

მენიუ

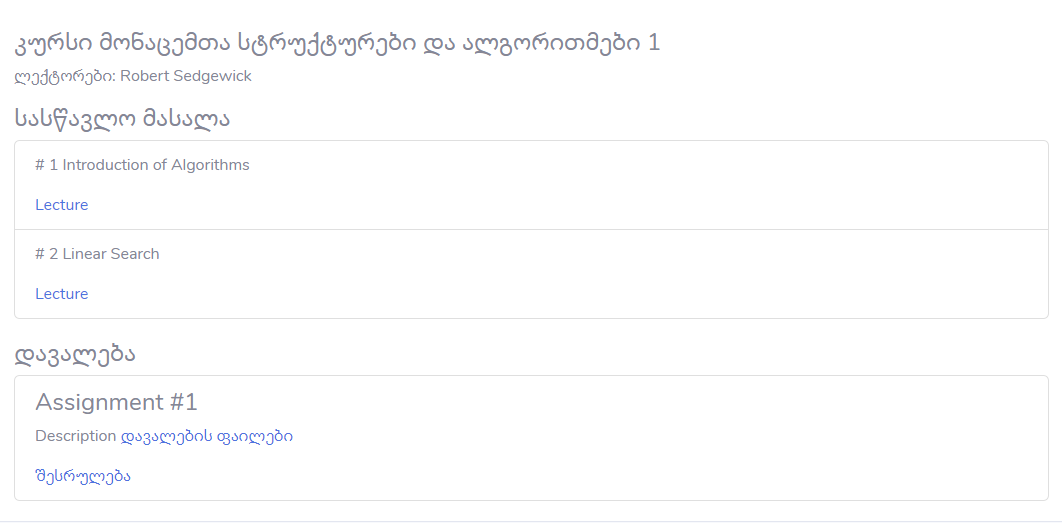
1. კურსები - კურსები, რომელსაც სტუდენტი სწავლობს
2. ენის ცვლილება - აპლიკაცია არის ორ ენოვანი: ინგლისური და ქართული

მთავარი ნაწილი - რომელ ლინკზეც გადავალთ მენიუდან, იმ ლინკის შესაბამისი ინფორმაცია ჩაიტვირთება ამ ნაწილში.

მომხმარებლის გამოსახულების კლავიშზე დაჭერისას ჩამოიშლება შემდეგი რამ: პარამეტრები და გამოსვლა.

1. პარამეტრები - მომხარებელს შეუძლია შეცვალოს თავისი ინფორმაცია: პაროლი, სახელი და ელ.ფოსტა.
2. გამოსვლა - მომხმარებლის გამოსვლა სისტემიდან.

სტუდენტის გვერდის კურსის დეტალური გვერდი



კურსის დეტალური გვერდის კომპონენტები:

1. კურსის დასახელება
2. ლექტორის მონაცემები
3. სასწავლო მასალა კვირის მიხედვით - კვირის თემა, ლინკები (სასწავლო მასალა ტექსტის, ვიდეოს ან აუდიო ფაილის სახით)
4. დავალებები - დავალების ნომერი, აღწერა და მიბმული ფაილი, სადაც უფრო ღრმად არის ახსნილი ან რა უნდა გაკეთდეს ან ამ ფაილში უნდა ჩავწეროთ ჩვენი შესრულებული დავალება
5. შესრულების ღილაკი - მასზე დაკლიკებისას გადავდივართ გვერდზე, სადაც შევძლებთ ავტვირთოთ ჩვენი საშინაო დავალება

**დასკვნა**

ჯგუფური პროექტის მსვლელობის მანძილზე ჩამოყალიბდა იდეა ელექტრონული სწავლების ვებ პორტალის შესახებ. სხვადასხვა კონსულტაციების, კვლევების შედეგად დადგინდა, გაირკვა პროექტის მოთხოვნები. იგი შედგებოდა შემდეგი მოდულებისგან: ლექტორის, სტუდენტის და ადმინისტრატორის გვერდებისგან. სამივე ტიპის მომხმარებელი ჰქმნის ერთიან სრულ პლატფორმას, მათ გარეშე შეუძლებელია იარსებოს ელექტორნულმა სწავლებამ. დავალებების ფუნქციონალი კიდევ უფრო გაზრდის სტუდენტების სწავლის ეფექტურობას. ისინი ისწავლიან ამა თუ იმ მასალას ყოველ კვირეულად, რადგან შეძლონ დაწერონ თითოეული კვირის მასალა. ეს მეთოდი სტუდენტებს ასწავლის დროის უკეთ მართვის, ორგანიზების უნარს.

პროექტის განსახორციელებლად გუნდის თითოელ წევრს გვქონდა დისკუსია ამა თუ იმ თემაზე, თუ რომელი ტექნოლოგია იქნებოდა უკეთესი ჩვენი ამოცანის გადასწყვეტად. საბოლოოდ ავირჩიეთ asp.net. ის არის ერთ-ერთი საუკეთესო საშუალება დიდი ზომის (Enterprise) აპლიკაციების შესაქმნელად. მონაცემთა ბაზად კი ავირჩიეთ მაიკროსოფტის Sql Server.

ჯგუფური პროექტის ფარგლებში შევძელით შეგვექმნა საიტის ინტერფეისი, მონაცემთა ბაზის არქიტექტურა და დემო ვებ აპლიკაცია.

**ლიტერატურა**

1. Professional Asp.Net Mvc 5 - Jon Galloway, Brad Wilson, K. Scott Allen, David Matson
2. N Tier Software Architecture, Jacob Jenkov - <http://tutorials.jenkov.com/software-architecture/n-tier-architecture.html>
3. Repository Design Pattern and Entity Framework <https://www.c-sharpcorner.com/UploadFile/8a67c0/repository-pattern-with-Asp-Net-mvc-with-entity-framework/>
4. დისტანციური სწავლება - <http://xelebiinga.blogspot.com/>
5. [Ben Pines - 7 UX Principles For Creating a Great Website](https://www.webdesignerdepot.com/2018/05/7-ux-principles-for-creating-a-great-website/)
6. [Alan Smith - 10 UX Tips For Your Next Website Redesign](https://usabilitygeek.com/10-ux-tips-for-your-next-website-redesign/)
7. Prados, F., Boada, I., Soler, J., Poch, J.: An automatic correction tool for relational database schemas Proc. of the 6th International Conference on Information Technology Based Higher Education and Training, 2005. ITHET 2005, S3C/9 - S3C14 (2005).
8. Felder, R. M. Felder, G. N., Dietz, E. J.: A longitudinal study of engineering student performance and retention. V. Comparisons with traditionally-taught students. Journal of Engineering Education, 87(4), 469-480 (1998).
9. Timmerman, B., Lingard, R.: Assessment of Active Learning with Upper Division Computer Science Students. Proc. of the 33rd ASEE/IEEE Frontiers in Education Conference, 2003, S1D-7-12 (2003).

**პროექტის სამუშაო გეგმა**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ამოცანები და აქტივობები** | **კვირები** | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **მონაცემთა ბაზის შექმნა** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| მოთხოვნებზე მსჯელობა |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| იმპლემენტაცია |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **სამომხმარებლო ინტერფეისი** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| დიზაინის შერჩევა |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| საწყისი გვერდის შექმნა |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| მომხმარებლის პირადი გვერდის შექმნა |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ადმინ პანელის შექმნა |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **პროგრამული კოდის დაწერა** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ბაზის დაკავშირება |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| საწყისი გვერდის ლოგიკა |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| მომხმარებლის პირადი გვერდის ლოგიკა |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ადმინისტრაციული პანელის ლოგიკა |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ტესტირება** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |