



ონლაინ სასწავლო ცხრილების პლატფორმა GTU TABLES

ამირან ქიმაძე¹; გულნარა კოტრიკაძე²

¹საქართველოს ტექნიკური უნივერსიტეტი, ინფორმატიკისა და მართვის სისტემების ფაკულტეტი, ბაკალავრიატის კურსდამთავრებული, უფროსი სპეციალისტი; ²საქართველოს ტექნიკური უნივერსიტეტი, ინფორმატიკისა და მართვის სისტემების ფაკულტეტი, ასოცირებული პროფესორი

აბსტრაქტი

ყოველთვის გვაწუხებდა ტექნიკური უნივერსიტეტის ახლანდელი ონლაინ ცხრილების პლატფორმა leqtori.gtu.ge მოუხერხებლობა, რადგან ის საკმაოდ მძიმედ მუშაობს და ძნელია სასწავლო ცხრილების მოძებნა, და დროულად ნახვა. გარდა ამისა მისი გამოყენება დღემდე ართულია. მისი სირთულე დამოკიდებულია მოწყობილობის, მობილურის, კომპიუტერის და ა.შ. ტექნიკურ მახასიათებლებზე, ოპერატიული მეხსიერების მოცულობაზე და თუნდაც ინტერნეტის სიჩქარეზე. ამ ყველაფრის ერთ-ერთ მთავარ მიზეზს წარმოადგენს ის, რომ ახლანდელი სასწავლო ცხრილების პლატფორმის შესაბამის გვერდებზე უნივერსიტეტის ყველა სასწავლო ჯგუფის, აუდიტორიისა თუ ლექტორების ცხრილები, ყველა ერთად იტვირთება, რაც ძალიან ართულებს სასურველი ცხრილის მოძიებას. იმისათვის, რომ ეს ყოველივე განვმატრო, გეტყვი, რომ მობილურით სარგებლობისას, საქართველოს ტექნიკური უნივერსიტეტის ონლაინ ცხრილები პლატფორმაზე სმარტფონით, რომელსაც 2GB ოპერატიული მეხსიერება გააჩნია და ვიყენებთ 4G ინტერნეტ კავშირს, სასურველი ჯგუფის სასწავლო ცხრილის მოძიება-გახსნისათვის საჭირო საშუალო დრო შეადგენს 1.7-დან 2 წუთამდე. საიდანაც გვერდის ჩატვირთვისათვის საჭირო დრო დაახლოებით 1.6 წამს შეადგენს. რაც სახარბიელო ნამდვილად არაა. უფრო მეტიც, თუ კი ტექნოლოგიების გიგანტ და დღეს მსოფლიოს ნომერ პირველ საძიებო სისტემა google-ს მიერ მოწოდებულ სტატისტიკას გავეცნობით, წავიკითხავთ რომ 2023 წლისთვის გვერდის ჩატვირთვის საშუალო დრო კომპიუტერისათვის შეადგენს 2.5 - 8.6 წამს შეადგენს, ხოლო მობილურზე ვებსაიტების საშუალო ჩატვირთვის დრო 3G კავშირზე 19 წამს შეადგენს.

გადავწყვიტეთ შეგვექმნა ონლაინ, სასწავლო ცხრილების ახალი სისტემა საქართველოს ტექნიკური უნივერსიტეტისათვის, რომელიც გააუმჯობესებს უნივერსიტეტის თითოეული პედაგოგისა და სტუდენტის საუნივერსიტეტო ცხოვრებას და ამავდროულად პრესტიჟს კიდევ უფრო აუმაღლებს ჩვენს პატივსაცემ, მრავალი პროფესიონალის აღმზრდელ და ძალიან დიდი

ისტორიის მქონე უნივერსიტეტს.

ამ და სხვა მოტივებიდან გამომდინარე, შევქმენით და წარმოგიდგინებ პროექტს, სახელწოდებით “GTU TABLES - სასწავლო ცხრილების პლატფორმა”.

კვლევა

იმ აზრის გასამყარებლად, რომ ნამდვილად არსებობდა ახალი სასწავლო ცხრილების ონლაინ პლატფორმის შექმნის აუცილებლობა, ჩავატარე გამოკითხვა. გამოკითხე 30 სტუდენტი. თითოეულ მათგანს დავუსვი შეკითხვები:

1. რამდენად კომფორტულია თქვენთვის სასწავლო ცხრილების ნახვა?

გამოკითხულთა 80%-მა პროცენტმა განაცხადა რომ ახლანდელი სასწავლო ცხრილების გვერდი არაკომფორტულია გამოსაყენებლად. 20 პროცენტი კი აფასებს ნეიტრალურად.

2. როგორ ნახულობთ სასწავლო ცხრილებს? ცხრილს ნახულობთ ყოველ ჯერზე ონლაინ თუ სარგებლობთ ერთჯერადად გადმოწერილი excel-ის ფაილით ან სქრინშოთით? აღნიშნულ შეკითხვაზე, გამოკითხულთა აბსოლიტურმა უმრავლესობამ აღნიშნა რომ სარგებლობს ერთჯერადად ჩამოტვირთული ექსელის ფაილით ან სქრინშოთით.

3. რა არის ამის მიზეზი (მეორე შეკითხვიდან გამომდინარე)?

გამოკითხულები უჩიოდნენ სასურველი სასწავლო ცხრილის სანახავად საჭირო დროის ხანგრძლივობას ნაწილმა აღნიშნა რომ მობილურით სარგებლობისას ცხრილების ნახვის დროს ზოგჯერ მობილური ჭედავს და ასევე გამოიკვეთა ის ფაქტიც რომ გამოკითხულთა ნაზილი ვერ ახერხებს გადმოწერილი excel-ის ფაილის მობილურში გახსნას.

4. ასევე დავსვი შეკითხვა, „რას შეცვლიდით არსებულ ონლაინ ცხრილების პლატფორმაზე?“

მივიღეთ პასუხები: „ყველაფერს“, „ახალია გასაკეთებელი, რაიმე თანამედროვე“, „იყოს რაც არის, მაინც იშვიათად შევდივარ“ და ა.შ.

გამოკითხვის შეჯამება: გამოკითხვის შედეგებმა, კიდევ ერთხელ დამარწმინა ჩემს მიერ ახალი ონლაინ ცხრილების პლატფორმის შექმნის საჭიროებაში.

ამოცანის დასმა

საერთო ჯამში, პროექტის მიზანია შეიქმნას უნივერსიტეტის სასწავლო ცხრილების ონლაინ პლატფორმა, სადაც მომხმარებლებს - სტუდენტებსა და პედაგოგებს შეეძლებათ გაეცნონ სასწავლო ჯგუფების, პედაგოგებისა და აუდიტორიული დატვირთვის ცხრილებს.

სასწავლო ცხრილების პლატფორმის აგება უნდა განხორციელდეს თანამედროვე მოთხოვნილებების შესაბამისად, UL/UX დიზაინის სატანარტების, საუკეთესო პრაქტიკების და თანამედროვე დიზაინ კონცეფციების გათვალისწინებით. იმგვარად რომ, დიზაინი იყოს

მარტივი, დახვეწილი, ინტუიტიური და ადვილად აღსაქმელი. რათა ვუზრუნველოყოთ დადებითი სამომხმარებლო გამოცდილება (User experience). საიტი უნდა იყოს მორგებული სხვადასხვა, გავრცელებულ მობილურ და კომპიუტერულ მოწყობილობებზე (responsive design).

ვებგვერდის ფუნქციონალური მოთხოვნები:

- სასწავლო ჯგუფების, პედაგოგების და აუდიტორიული ცხრილების ძებნა და გაცნობა;
- სასწავლო განრიგის ნახვა საათებისა და დღეების მიხედვით;
- იმ ჯგუფების სიისა და ცხრილების სიის ნახვა, რომლებსაც ლექცია თუ ლექციები უტარდებათ მასთან და მის ჯგუფთან ერთად;
- მომხმარებელს უნდა შეეძლოს ნახოს, თუ როდის იწყება და მთავრდება ლექცია;
- მომხმარებელს უნდა შეეძლოს ნახვა იმისა, თუ ვინ ატარებს ლექციას;
- პედაგოგმა უნდა შეეძლოს მისი ცხრილის ძებნა და მასთან გაცნობა;
- პედაგოგს უნდა შეეძლოს იმ ჯგუფების სიის ნახვა რომლებსაც ის უტარებს ლექციას ამა თუ იმ დროს;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს, გაეცნოს სტატისტიკას შემდეგი სახის ჩანაწერების რაოდენობის შესახებ: კორპუსები, აუდიტორიები, ჯგუფები, ლექტორები, სასწავლო საგნები, მომხმარებლები (ადმინისტრატორები), სოც. ბმულები და ა.შ.;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს სამართავ პანელში შესვლა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს სამართავ პანელში უნივერსიტეტის კორპუსების შესახებ არსებული ჩანაწერების მენეჯმენტი, დამატება, რედაქტირება და ამოშლა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ნახოს, გაეცნოს კორპუსებში აუდიტორიების რაოდენობას;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს კორპუსებში ოთახების რაოდენობისა და სიის ნახვა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს კორპუსების შესახებ ჩანაწერებთან დაკავშირებული აუდიტორიების შესახებ არსებული ჩანაწერების, ოთახების სიის მენეჯმენტის განხორცილება. დამატება, რედაქტირება და წაშლა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ოთახების ძიება, ფილტრაცია ოთახის დასახელებითა და კორპუსის ნომრით;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ოთახების სიის იმპორტირება excel-ის ფაილიდან მონაცემთა ბაზაში;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს სემესტრების შესახებ ჩანაწერების დამატება, რედაქტირება და წაშლა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ამა-თუ-იმ სემესტრში ლექციების საერთო რაოდენობის ნახვა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს სემესტრი მონიშნოს, აირჩიოს მიმდინარედ;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს, აირჩიოს სასურველი სემესტრი და

მოახდნოს სალექციოს ცხრილების იმპორტირება excel-ის ფილიდან მონაცემთა ბაზაში, არჩეულ სემესტრში;

- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს სწარავი ნავიგაცია სასწავლო ცხრილების იმპორტირების/ატვირთვის გვერდიდან ჯგუფების სიის გვერდზე და სასურველი ჯგუფის სასწავლო ცხრილებზე;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ლექტორების სიის ნახვა, ლექტორების სიის მენეჯმენტის განხორციელება, დამატება, რედაქტირება და წაშლა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს დააწესოს შეზღუდვა დღეებისა და საათების მიხედვით თითოეული ლექტორის შემთხვევაში ინდივიდუალურად. თუ ლექტორს არ სცალია კონკრეტულ საათსა თუ კვირის რომელიმე დღეს. ადმინისტრატორს უნდა შეეძლოს მონიშნოს კონკრეტული დღე ან საათი „დაკავებულ საათად, საათებად თუ დღეებად“;
- ლექტორის დამატების შემდეგ, შესაძლებელი უნდა იყოს ლექტორისთვის საგნის ჩასმა;
- ლექტორისთვის საგნის ჩასმისას ვებსაიტის ადმინისტრატორს უნდა შეეძლოს, სასურველი სემესტრის, ჯგუფის, რამდენიმე ჯგუფის ერთად მონიშვნა, სასწავლო საგნების, ლექციის ხანგრძლივობის, ხელსაყრელი კორპუსის არჩევა, რემდენიმე კორპუსის ერთად მითითება, ოთახის კოდის მითითება;
- წინათ ჩამოთვლილ ველებს თან უნდა ახლდეს autocomplete, suggestions რათა ვებსაიტის ადმინისტრატორს გაუადვილდეს ველების შევსება;
- საგნის ჩასმა უნდა მოხდეს ავტომატურად, ლექციის ჩამატების დროდ და დღეთ კი უნდა განისაზღვროს ყველაზე ადრეული დღე და საათი;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ლექტორებისთვის ჯგუფების მიმაგრება და მიმაგრებული ჯგუფების სიის ცვლილება;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ლექტორების სიის ძებნა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ჯგუფების სიის ნახვა და მენეჯმენტი, დამატება, რედაქტირება და წაშლა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს სასწავლო ჯგუფების ძებნა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს კონკრეტული ჯგუფების ცხრილების ნახვა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს სასწავლო ჯგუფის ცხრილების ნახვა სემესტრების მიხედვით;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს სასწავლო ჯგუფის ნებისმიერ ცხრილში ლექციის ჩამატება;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს, კონკრეტული სასწავლო ჯგუფისათვის ლექციის ჩამატების ფორმაში შემდეგი ველების შევსება ან არცევა: სემესტრი, ლექციის დასახელება, ლექტორი, ხანგრძლივობა, კორპუსი, ოთახი;
- ლექციისათვის დრო და აუდიტორიის არ არჩევის შემთხვევაში, თავისუფალი დროც და აუდიტორიაც უნდა გამოინახოს ავტომატურად, იმგვარად რომ ლექცია ჩაინიშნოს რაც შეიძლება ადრე კვირის განმავლობაში;
- საიტის ადმინისტრატორს უნდა შეეძლოს ვებსაიტის სხვადასხვა საკონტაქტო

ინფორმაციის რედაქტირება;

- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს უნივერსიტეტის სოციალური ქსელების ბმულების ჩამატება, რედაქტირებამ გადანაცვლება პოზიციის და წაშლა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს SEO (Search Engine Optimisation) პარამეტრების, მეტა ტეგების რედაქტირება-შეცვლა პირდაპირ სამართავი პანელიდან;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს ლოადერსი ჩართვა და გამორთვა;
- ვებსაიტის ადმინისტრატორს უნდა შეეძლოს პირადი და სხვა ადმინისტრატორების მონაცემების რედაქტირება.

თავი 1

ამოცანის გადაწყვეტის გზები

ნებისმიერი ვებსაიტის აგებისას მნიშვნელოვანია, რომ ერთ-ერთი მთავარი აქცენტი გაკეთდეს დიზაინზე, რაც არ უნდა სამწუხარო იყოს ჩემთვის და ჩემი Backend დეველოპერი კოლეგებისთვის, მომხმარებელი ხედავს და აღიქვავს მხოლოდ დიზაინს, არა იმ კოდს, იმ შრომას, რომელიც დგას ლამაზი დიზაინის უკან, უშუალოდ ვებსაიტის ფუნქციონირებისათვის საჭირო პროგრამული ალგორითმებისა და სხვა ლოგიკური კოდის სახით. ამიტომაც საჭიროა, რომ დიზაინი ჩამოვყალიბოთ იმგვარად, რომ ყველაფერი ერთმანეთთან ლოგიკურად იყოს დაკავშირებული.

კარგი დიზაინის შესაქმნელად პირველ რიგში სასურველია დიზაინის მონახაზის გაკეთება ისეთი ხელსაწყოებით, რომელიც საშუალებას მოგვცემს, რომ დახატული ვებ დიზაინი გარდავქმნათ მარტივად და მარტივად მივამსგავსოთ რეალურ შესაძლო ვარიანტს.

რაც შეეხება პროგრამული ენებისა და ფრეიმვორქების გამოყენების ნაწილს, აუცილებელია, რომ ტექნოლოგიების არჩევისას გავითვალისწინოთ მრავალი ფაქტორი. უპირველეს ყოვლისა, რათქმაუნდა, ვითვალისწინებთ იმას, თუ რამდენად კარგად ვფლობთ ამა თუ იმ ტექნოლოგიას (პროგრამულ ენას, ფრეიმვორქს), ვითვალისწინებთ, თუ რომელი ტექნოლოგია დღეს დღეისობით გამოყენების დანიშნულებისამებრ ყველაზე აქტუალურ დღევანდელ ბაზარზე. რომელი მათგანია ყველაზე განახლებადი, ხელმისაწვდომი ამავედროულად მოქნილი.

სასურველია, რომ საიტის დიზაინს უკან ამოფარებული ლიგოკა სრულდებოდეს სერვერის და არა მომხმარებლის მხარეს. რაც იძლევა მეტ უსაფრთხოებას და ვებგვერდის შედარებით სწრაფი რენდერირების საშუალებას. ასევე, სერვერის მხარეს რენდერირება (Server side rendering) იძლევა უკეთეს შედეგს საძიებო სისტემებთან ოპტიმიზაციის თემაში. საძიებო სისტემებთან ოპტიმიზაცია გულისხმობს იმას, რომ როდესაც მომხმარებელი ეძებს მაგალითისათვის, სასწავლო ცხრილებს გუგლში, საჭიროა რომ ჩვენს მიერ შექმნილი ვებსაიტი გამოდიოდეს ყოველთვის ძიების შედეგების თავში. თუმცა ეს

ამავდროულად არ ნიშნავს იმას, რომ თუ მაგალითად ვებსაიტს შევქმნით php ენაზე, რომ ჩვენი ვებსაიტი აუცილებლად ძიების შედეგების თავში მოხვდება. იმისათვის რომ ჩვენი ვებსაიტი გუგლის ძიების შედეგების თავში მოხვდეს, ამისთვის SEO სპეციალისტების კარგი მუშაობა საჭირო. SEO ფაქტობრივად ცალკე „ხელოვნება“, რომელსაც შესაბამისი დარჯის ქურუმები ფლობენ.

აქვე, არ უნდა დაგვავიწყდეს მთავარი, თუ რა გვიჯდება თვიურად ტექნოლოგიებზე აგებული ვებსაიტის დაჰოსტვა. მაგალითისათვის: დღეს დღეობით, ქართულ ბაზარზე საქართველოში ერთ-ერთ ყველაზე პოპულარულ ჰოსტინგ პროვაიდერ ProService.ge-ზე, C#, ASP.NET-ზე დაწერილი ვებსაიტის დაჰოსტვა (სერვერის ქირაობა) 5GB სივრცით, 55 ლარი ჯდება. ხოლო php ენაზე დაწერილი ვებსაიტის დაჰოსტვა 5GB სივრცით საქართველოში ნომერ პირველ ჰოსტინგ პროვაიდერ ProServ.ge-ზე წინა ვარიანტთან შედარებით სულ რაღაც 7 ლარი ჯდება თვიურად.

ტექნოლოგიების არჩევისას ვებ საიტების შექმნის მომენტში უნდა გავითვალისწინოთ მოსალაოდნელი ვიზიტორთა რაოდენობაც და ტრაფიკი რომელის მიღებასაც უნდა ველოდეთ ვებსაიტის აქტიურ ხმარებაში ჩაშვების შემთხვევაში. ჩვენს მიერ შექმნილმა ვებგვერდმა უნდა გაუძლოს ამ ყველაფერს. სერვერმა კი დაამუშაოს ჩვენი ლოგიკური პროგრამული კოდი და შეარულოს თითოეული მომხმარებლის მიერ განხორციელებილი მოთხოვნა.

რაც ყველაზე მთავარია, ყველა იმ ხელსაწყოთა, ტექნოლოგიათა ერთობლიობა ჯემში უნდა გვამღებდეს რეალისტურ შესაძლებლობას რომ ვუპასუხოთ იმ გამოწვევებს რომელიც დავისახეთ და მათაც რომლებიც შეიძლება გაგვიჩნდეს მომავალში.

ამ ყველაფრიდან გამომდინარე აღნიშნული პროექტისთვის შევარჩიეთ ტექნოლოგიები და ხელსაწყოები, რომელთა გამოყენება, ამ კონკრეტულ შემთხვევაში ყველაზე მეტად მაწყობდა.

1.1. გამოყენებული ხელსაწყოები და იმპლემენტაციის დეტალები

პროექტი შექმნილია შემდეგი ტექნოლოგიებით და ხელსაწყოებით:

- HTML (Markup language)
- CSS (ვებსაიტის ვიზუალიზაციის ენა)
- Bootstrap 5 (CSS Framework)
- JQuery (JavaScript ბიბლიოთეკა),
- Laravel PHP Framework
- MySQL მონაცემთა ბაზა
- Line awesome (Font Icons)
- Lunacy (Designing software)
- Paint.NET - გრაფიკულ გამოსახულებასთან სამუშაო software

HTML (The HyperText Markup) სტანდარტული მარჯაფის ენა ვებსაიტის ვიზუალური მხარის ასაგებად რომელსაც იერსახეს აძევს მასტან ერთად CSS ენის გამოყენება.

CSS (Cascading Style Sheets) არის ტექნოლოგია რომელიც გამოიყენება HTML დოკუმენტის გარე იერსახის აღწერისათვის, სტილიზაციისათვის.

Bootstrap არის ღია წყაროს ფრონტ-ენდის ჩარჩო, რომელიც დეველოპერებს საშუალებას აძლევს სწრაფად შექმნან მოწყობილობაზე მორგებადი (responsive) და ვიზუალურად მიმზიდველი ვებსაიტები. ის უზრუნველყოფს CSS და JavaScript კომპონენტების კრებულს, როგორცაა ღილაკები, ფორმები, სანავიგაციო ზოლები, მოდალები და სხვა, მოწყობილობაზე მორგებადი (responsive) Grid სისტემასთან ერთად, რომელიც ხელს უწყობს დინამიური და მობილურისთვის მოსახერხებელი ელემენტთა განლაგების შექმნას.

ახლა მოდით გადავიდეთ Bootstrap დადებით ასპექტებზე:

1. ოპტიმიზაციას მობილური მოწყობილობებისთვის. ეს სტრატეგია უზრუნველყოფს, რომ Bootstrap-ით შექმნილი საიტები იყოს მობილურისთვის მოსახერხებელი და ინტერაქტიული, რაც ემსახურება მობილური მომხმარებლების მზარდ რაოდენობას.
2. დროის ეფექტურობა: Bootstrap-ის წინასწარ ჩაშენებული კომპონენტები და მოწყობილობაზე მორგებადი (responsive) სისტემა აჩქარებს პროექტის შექმნის პროცესს, რაც დეველოპერებს საშუალებას აძლევს შექმნან პროტოტიპები ან დაასრულონ ვებსაიტები სწრაფად. რაც განსაკუთრებით სასარგებლოა მჭიდრო ვადების მქონე პროექტებისთვის.
3. ინტეგრაცია და გაფართოება: Bootstrap შეუფერხებლად ინტეგრირდება სხვა ბიბლიოთეკებთან და ჩარჩოებთან, როგორცაა jQuery, რაც დეველოპერებს საშუალებას აძლევს გამოიყენონ დამატებითი ფუნქციები და გააუმჯობესონ თავიანთი პროექტები. უფრო მეტიც, Bootstrap გთავაზობთ მესამე მხარის (Third party themes) თემების, შაბლონების და დანამატების ფართო სპექტრს, რაც კიდევ უფრო აფართოებს მის შესაძლებლობებს.

jQuery არის პოპულარული და ლეგენდარული JavaScript ბიბლიოთეკა. ის სწრაფია, მცირე ზომის და მრავალფუნქციური. იგი ამარტივებს მუშაობას მოვლენებთან, სელექტორებთან, ანიმაციებთან, API-სთან, ზოგადად იგი ამარტივებს ყველაფერს და ამოკლებს ჯავასკრიპტის სინტაქსს რაც საშუალებას გვაძლევს რომ ნაკლები კოდის დაწერის ხარჯზე მოვიგოთ დრო და გავაკეთოთ მეტი შეზღუდულ დროში.

Bootstrap არის ღია წყაროს ფრონტ-ენდის ჩარჩო, რომელიც დეველოპერებს საშუალებას აძლევს სწრაფად შექმნან მოწყობილობაზე

Laravel დღეს დღეისობით საკმაოდ ფართოდ გამოიყენება Model-View-Controller (MVC) არქიტექტურის PHP framework-ს წარმოადგენს. რომელიც გვებმარება ვებ

დეველოპერებს რომ გავამარტივოთ ვებ აპლიკაციების შექმნის პროცესი. მისი გამოყენებით ვებ ალიკაციის კოდი ხდება მეტად განიზიარებული და მარტივად რომ ავხსნა - მივეყვებით კოდის წერის გარკვეულ სტილს და ყველა კოდს მისი დანიშნულებიდან გამომდინარე ვახარისხებთ/ვწერთ შესაბამის ფაილებსა და საქალაქდებში. ამგვარად ისეთ შემთხვევაში პროექტზე მუშაობს დეველოპერთა გუნდი, თითოეულს ეცოდინება თუ სად რა კოდი უნდა ეწეროს და რომელი ფაილი სად არის კონცენტრირებული. ლარაველთან ერთად ვებ დეველოპერებს შეგვიძლია ვწერთ კოდუ უფრო სუფთად (clean coding) და ისე რომ ჩვენი კოდი იყოს უფრო იოლიად წაკითხვადი, აღსაქმელი. რადგანაც ლარაველს გააჩნია მომხმარებელზე მორგებული სინტაქსი. იგი ამატრივებს მუშაობას მონაცემთა ბაზებთან, routing-თან, caching-თან და მეტად უსაბრთხოს ხდის მომხმარებლის ავტორიზაციას. Laravel-ს ასევე გააჩნია ჩაშენებული „templating engine“ რომელიც ამარტივებს templating-ს. აღნიშნულ თემფლეითინგ ენჯინს „Blade“ ეწოდება. ლარაველს ასევე გააჩნია უზარმაზარი საზოგადოება/community რომელიც გვთავაზობს განახლებებს, ვრცელ ეკოსისტემას, სხვადასხვა გაფართოებებს ლარაველისათვის და ა.შ.

MySQL არის ფართოდ გამოყენებული და სანდო ღია კოდის მონაცემთა ბაზის მართვის სისტემა. იგი უპირატესობას შესრულების სისწრაფეს, გამოყენების სიმარტივის, მოქნილობას, უსაფრთხოების, ხარჯების ეფექტურობისა და ძლიერი community support-ს. MySQL ცნობილია თავისი სტაბილურობით, მონაცემთა ბაზის მოთხოვნების ეფექტური შესრულებით და დიდი რაოდენობით მონაცემების დამუშავების უნარით. ის გთავაზობს სხვადასხვა ინსტრუმენტებს და მხარს უჭერს მრავალ პროგრამირების ენას. გარდა ამისა, MySQL უზრუნველყოფს უსაფრთხოების მძლავრ მახასიათებლებს და მისი გამოყენება უფასოა, რაც მას შესანიშნავ არჩევანს ხდის აპლიკაციების ფართო სპექტრისთვის.

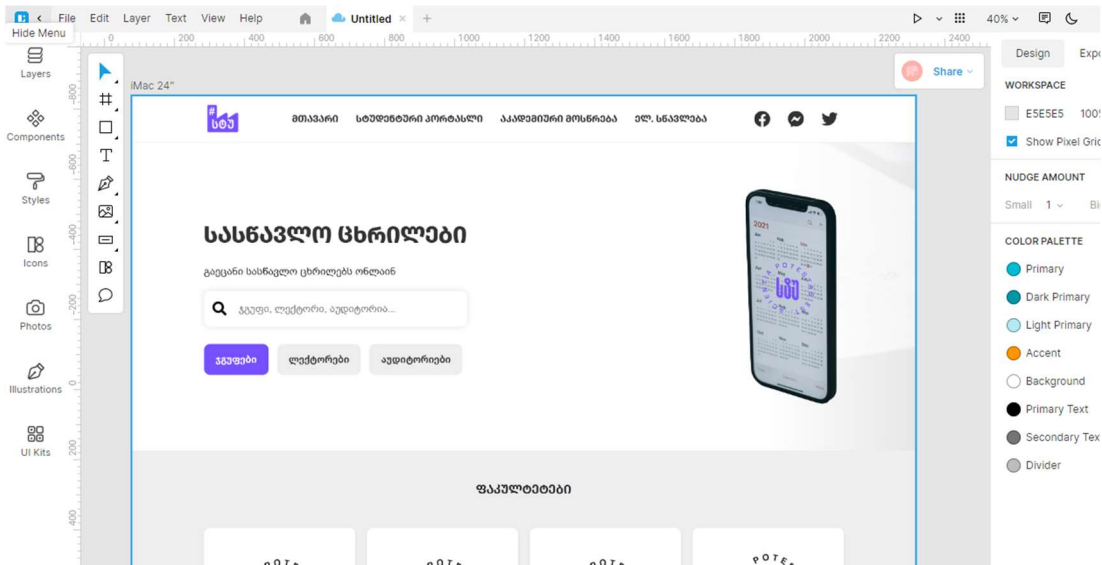
Line Awesome ფახლავთ css ბიბლიოთეკა Icons პაკეტი რომელიც გთავაზობს მეტად ლამაზ ფონტურ Icon-ს. რომელთა გამოყენება საკმაოდ ეფექტურია და დიზაინს უფრო ეფექტურს ხდის. მას გააჩნია გამზადებული კლასები რომელთა საშუალებითაც გამოგვაქვს icon-ები. მისი ვებ პროექტში ინტეგრაცია ძალიან ადვილია და მოიხმარს საკმაოდ მცირე რესურსს რაც არ ანელებს ვებსაიტის ჩატვირთვის პროცესს.

Lunacy არის საკმაოდ მოსახერხებელი და უასო დიზაინ პროგრამული უზრუნველყოფა რომელიც საშუალებას გვაძლევს გავაკეთოს ვებსაიტის მონაზი მის უშალოგებამდე. ის საშუალებას გვაძლევს რომ დავხატოთ ვებსაიტის სხვადასხვა ვარიაციები, შევიტანოთ დიზაინში ცვლილებები და ა.შ.

Paint.NET არის პოპულარული და მოსახერხებელი გამოსახულების რედაქტირების პროგრამა Windows-ისთვის. იგი საშუალებას გვაძლევს მოვახდინოთ ფოტოების, ლოგოების მონტაჟი და დამუშავება გარვეულ დონეზე. იგი გვაძლევს ფოტოშოპის მსგავს შესაძლებლობებს, თუმცა იგი არ გამოიყენება როგორც პროფესიონალური, გრაფიკულ გამოსახულებაზე სამუშაო პროგრამა. თუმცა საკმაოდ ბევრი რამის გაკეთების საშუალებას იძლევა.

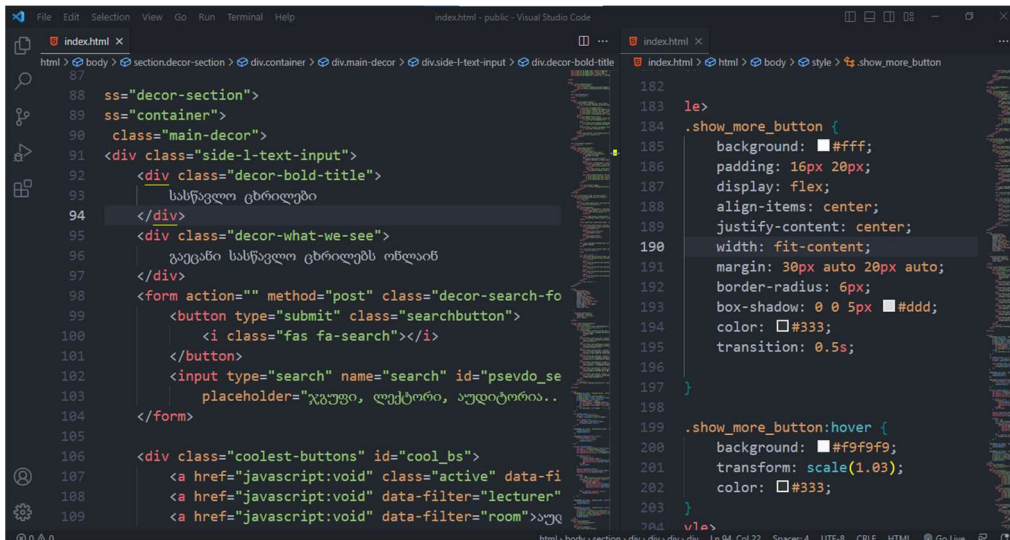
1.2.პროექტის შექმნის პროცესი

პროექტის შექმნა დავიწყეთ ვებსაიტის დიზაინის შემუშავებით პროგრამაში Lunacy. რომელმაც გამიმარტივა ვებგვერდის დიზაინის შემუშავება, დიზაინში გამოყენებული კომპონენტების მდებარეობის, ფერების, ფორმების, ცხრილების და ა.შ.



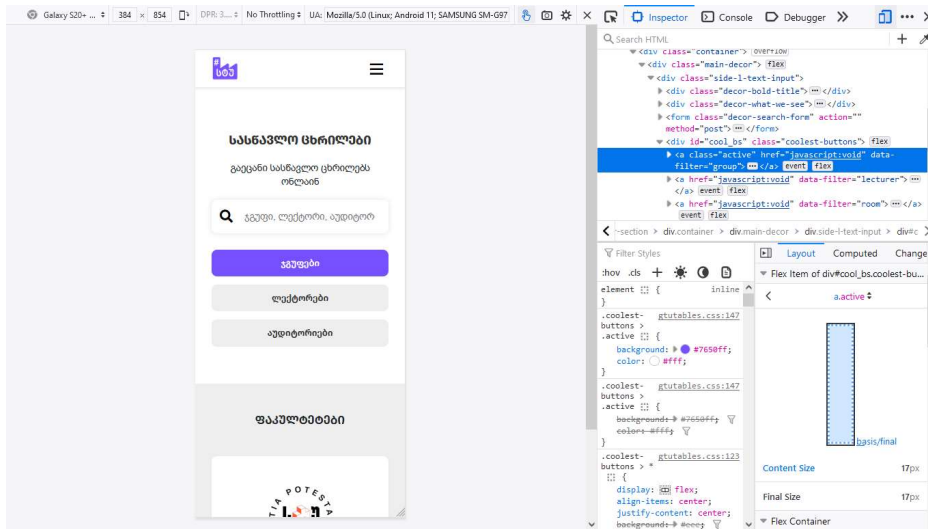
სურათი 1.1. დიზაინის მონახაზი

HTML, CSS, Bootstrap 5 და JQuery საშუალებით ავაგე ვებსაიტის მაკეტი. ვებსაიტის Front-end მხარე:



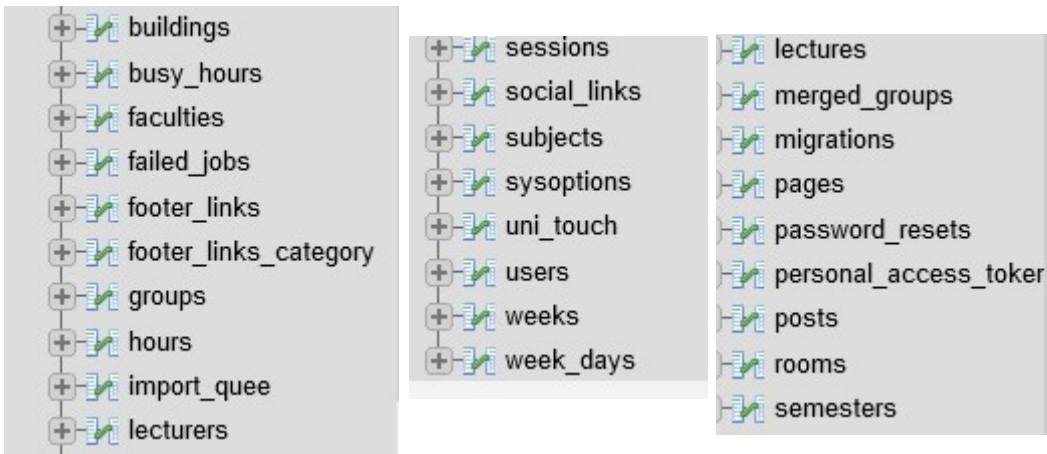
სურათი 1.2. დიზაინის კოდის ფრაგმენტი

შემდგომ მოვახდინეთ შექმნილი დიზაინის მორგება სხვადასხვა მოწყობილობებზე, მობილურზე, პლანშეტზე, ლეპტოპზე და ა.შ. შევითანეთ მცირედი კორექციები დეფექტების აღმოსაფხვრელად.



სურათი 1.3. დიზაინის დივასებზე მორგება

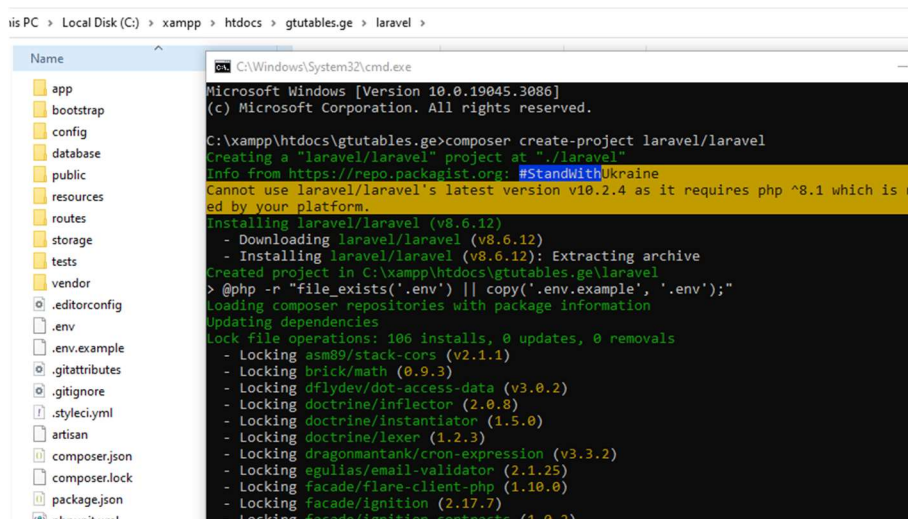
დიზაინის დამუშავების შემდეგ, არსებული მოთხოვნებიდან გამომდინარე, დავიწყეთ მონაცემთა ბაზის ჩამოყალიბება და შევქმენი შემდეგი ცხრილები:



სურათი 1.4. მონაცემთა ბაზების ცხრილები

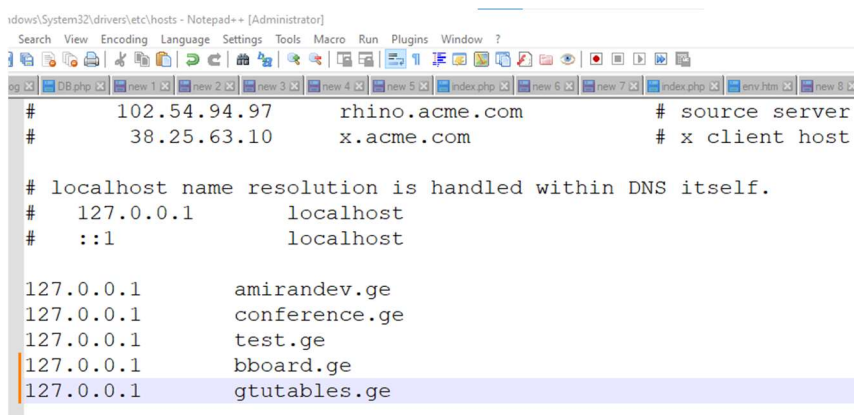
გავხსენი Command Prompt შესაბამის საჯალდეში და დავიწყე Laravel Framework-ის დაყენება. ამისათვის კომანდლაინში გავუშვი ბრძანება:

composer create-project laravel/laravel



სურათი 1.5. laravel-ის გადმოწერა

Laravel Framework-ის გადმოწერის შემდეგ, მუშაობის გასამარტივებლად ჩემს კომპიუტერზე ჩავამატე ვირტუალური ჰოსტი. რის გამოც მომიწია C:\Windows\System32\drivers\etc\hosts გახსნა code რედაქტორში ადმინისტრატორის უფლებით რათა ჩამემატებინა ჩემი ვირტუალური დომენი. (ფოტოზე ასევე დაინახავთ ჩემს მიერ ჩამატებულ სხვა ვირტუალურ დომენებსაც რომელიც ამ კონკრეტულ პროექტთან არაა ასოცირებული)



სურათი 1.6. ვირტუალური დომენები

ვირტუალური დომენის ჩამატების შემდეგ გავხსენი ჩემი ვირტუალური სერვერის დომენური კონფიგურაციის ფაილი **httpd-vhosts.conf** საქალაქო C:\xampp\apache\conf\extra და მოვახდინე ვირტუალური ჰოსტის კონფიგურირება

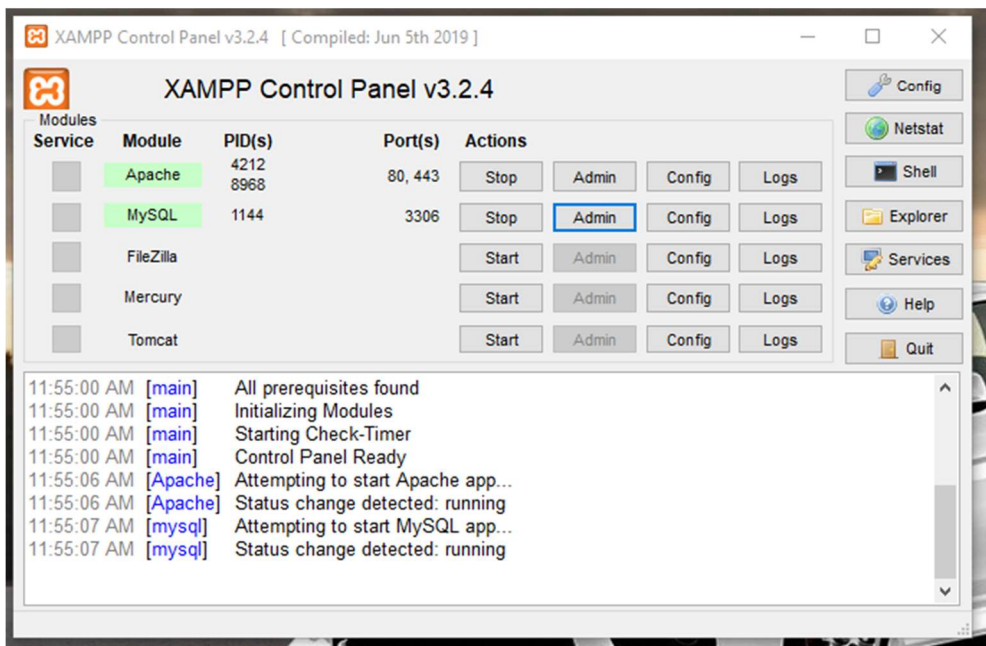
```

<VirtualHost *:80>
    ServerAdmin webmaster@gtutables.ge
    DocumentRoot "C:/xampp/htdocs/gtutables.ge/public"
    ServerName gtutables.ge
    ErrorLog "logs/gtutables-error.log"
    CustomLog "logs/gtutables.log" common
</VirtualHost>

```

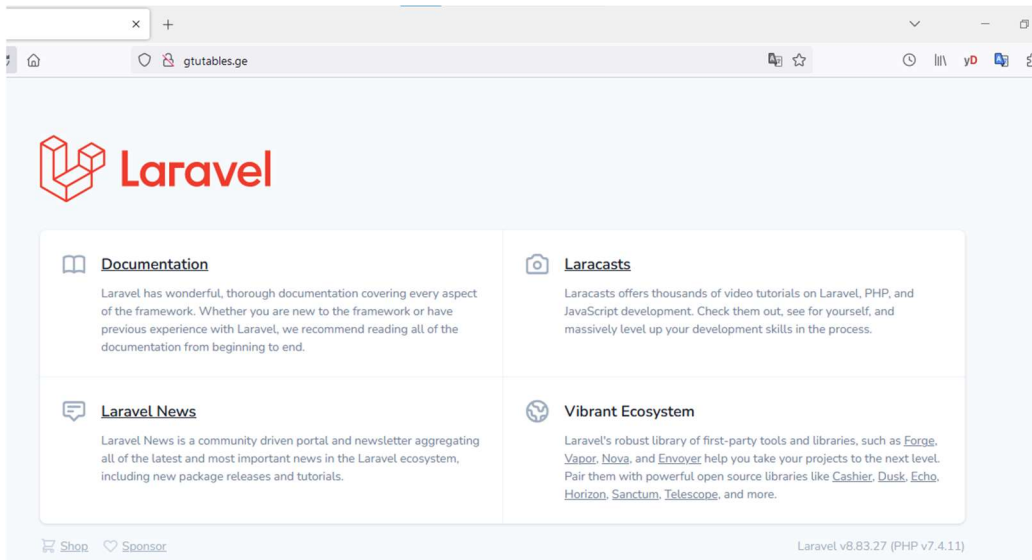
სურათი 1.7. ვირტ. ჰოსტის კონფიგურაცია

ამ ყველაფრის შემდეგ დავსტარტე ჩემი ლოკალური სერვერი



სურათი 1.8. xampp control panel

გავხსენი ჩემს მიერ ჩამატებული ვირტუალური დომენი ბრაუზერში და შევეგებე ცარიელ/ახლად დაყენებულ ლარაველ აპლიკაციას. რომლის სურათიც შეგიძლიათ იხილოთ ქვემოთ



სურათი 1.9. responsive design

შემდგომ გავხსენით პროექტი Code რედაქტორ VS Code-ში და მასში ჩაშენებული ტერმინალის გამოყენებით შევქმენი მონაცემთა ბაზების ცხრილების მოდელები „**php artisan make:model** დასახელება“.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\xampp\htdocs\gtutables.ge> php artisan make:model Users
Model created successfully.
PS C:\xampp\htdocs\gtutables.ge> php artisan make:model Rooms
Model created successfully.
PS C:\xampp\htdocs\gtutables.ge> php artisan make:model Buildings
Model created successfully.
PS C:\xampp\htdocs\gtutables.ge> php artisan make:model BusyHours
Model created successfully.
PS C:\xampp\htdocs\gtutables.ge> php artisan make:model MergedGroups
Model created successfully.
PS C:\xampp\htdocs\gtutables.ge>
  
```

სურათი 1.10. DB models

მონაცემთა ბაზის შესაბამისი მოდელების შექმნის შემდეგ, შევქმენით კონტროლერები კონტროლერებიც, შემდეგი ბრძანების გამოყენებით: **php artisan make:controller** კონტროლერის სახელი

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\xampp\htdocs\gtutables.ge> php artisan make:controller Homecontroller
Controller created successfully.
PS C:\xampp\htdocs\gtutables.ge> php artisan make:controller GroupsController
Controller created successfully.
PS C:\xampp\htdocs\gtutables.ge> php artisan make:controller LecturesController
Controller created successfully.
PS C:\xampp\htdocs\gtutables.ge> php artisan make:controller LecturersController
Controller created successfully.
PS C:\xampp\htdocs\gtutables.ge>
  
```

სურათი 1.11. responsive design

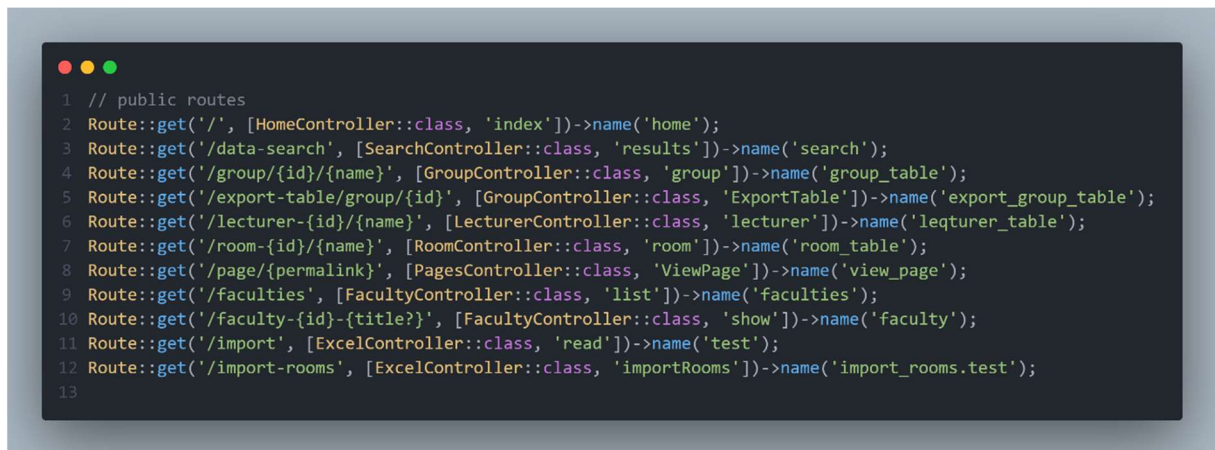
შევქმენით შესაბამისი დიზაინის ფაილები ლარაველ პროექტისთვის blade.php გაფართოებით



```
1 @extends('layouts.default')
2 @section('content')
3     <section class="decor-section">
4         <div class="container">
5             <div class="main-decor">
6                 <div class="side-l-text-input">
7                     <div class="decor-bold-title">
8                         სასწავლო ცხრილები
9                     </div>
10                    <div class="decor-what-we-see">
11                        გაეცანი სასწავლო ცხრილებს ონლაინ
12                    </div>
13                    <form action="" method="post" class="dec
14                        <button type="submit" class="searchb
```

სურათი 1.12. blade ნიმუში

შემდეგ, შევქმენი როუტები მომხმარებლის მხარისათვის



```
1 // public routes
2 Route::get('/', [HomeController::class, 'index']->name('home'));
3 Route::get('/data-search', [SearchController::class, 'results']->name('search');
4 Route::get('/group/{id}/{name}', [GroupController::class, 'group']->name('group_table');
5 Route::get('/export-table/group/{id}', [GroupController::class, 'ExportTable']->name('export_group_table');
6 Route::get('/lecturer-{id}/{name}', [LecturerController::class, 'lecturer']->name('lecturer_table');
7 Route::get('/room-{id}/{name}', [RoomController::class, 'room']->name('room_table');
8 Route::get('/page/{permalink}', [PagesController::class, 'ViewPage']->name('view_page');
9 Route::get('/faculties', [FacultyController::class, 'list']->name('faculties');
10 Route::get('/faculty-{id}-{title?}', [FacultyController::class, 'show']->name('faculty');
11 Route::get('/import', [ExcelController::class, 'read']->name('test');
12 Route::get('/import-rooms', [ExcelController::class, 'importRooms']->name('import_rooms.test');
13
```

სურათი 1.13. public routes

ასევე, შევქმენი როუტების ნაწილი ადმინისტრატორის მხარისათვის

```

1
2 Route::prefix('admin')->middleware(['auth'])->group(function () {
3     Route::get('/', [DashboardController::class, 'Dashboard'])->name('dashboard');
4
5     // define building routes
6     Route::prefix('semesters')->group(function () {
7         Route::get('/', [SemesterController::class, 'List'])->name('admin_semesters');
8         Route::get('/data.json', [SemesterController::class, 'Json'])->name('admin_semesters.json');
9         Route::post('/post-add', [SemesterController::class, 'Add'])->name('admin_semesters.post_add');
10        Route::post('/post-edit', [SemesterController::class, 'Edit'])->name('admin_semesters.post_edit');
11        Route::post('/post-delete', [SemesterController::class, 'Delete'])->name('admin_semesters.post_delete');
12        Route::get('/search-semesters.json', [SemesterController::class, 'searchJson'])->name('admin_semesters.search_json');
13    });
14
15    // define building routes
16    Route::prefix('buildings')->group(function () {
17        Route::get('/', [BuildingsAdminController::class, 'List'])->name('admin_buildings');
18        Route::get('/data.json', [BuildingsAdminController::class, 'Json'])->name('admin_buildings.json');
19        Route::post('/post-add', [BuildingsAdminController::class, 'Add'])->name('admin_buildings.post_add');
20        Route::post('/post-edit', [BuildingsAdminController::class, 'Edit'])->name('admin_buildings.post_edit');
21        Route::post('/post-delete', [BuildingsAdminController::class, 'Delete'])->name('admin_buildings.post_delete');
22        Route::get('/search-buildings.json', [BuildingsAdminController::class, 'searchJson'])->name('admin_buildings.search_json');
23    });
24
25    Route::prefix('rooms')->group(function () {
26        Route::get('/', [RoomsAdminController::class, 'List'])->name('admin_rooms');
27        Route::get('/data.json', [RoomsAdminController::class, 'Json'])->name('admin_rooms.json');
28        Route::post('/upload', [RoomsAdminController::class, 'upload'])->name('admin_rooms.upload');
29        Route::get('/run-import/{id}', [RoomsAdminController::class, 'runImport'])->name('admin_rooms.run_import');
30        Route::post('/post-add', [RoomsAdminController::class, 'Add'])->name('admin_rooms.post_add');
31        Route::post('/post-edit', [RoomsAdminController::class, 'Edit'])->name('admin_rooms.post_edit');
32        Route::post('/post-delete', [RoomsAdminController::class, 'Delete'])->name('admin_rooms.post_delete');
33        Route::get('/search-rooms.json', [RoomsAdminController::class, 'searchJson'])->name('admin_rooms.search_json');
34    });
35
36    Route::prefix('groups')->group(function () {
37        Route::get('/', [GroupsAdminController::class, 'List'])->name('admin_groups');
38        Route::get('/data.json', [GroupsAdminController::class, 'Json'])->name('admin_groups.json');
39        Route::post('/upload', [GroupsAdminController::class, 'upload'])->name('admin_groups.upload');
40        Route::get('/run-import/{id}', [GroupsAdminController::class, 'runImport'])->name('admin_groups.run_import');
41        Route::post('/post-add', [GroupsAdminController::class, 'Add'])->name('admin_groups.post_add');
42        Route::post('/post-edit', [GroupsAdminController::class, 'Edit'])->name('admin_groups.post_edit');
43        Route::post('/post-delete', [GroupsAdminController::class, 'Delete'])->name('admin_groups.post_delete');
44        Route::get('/search-groups.json', [GroupsAdminController::class, 'searchJson'])->name('admin_groups.search_json');
45    });

```

სურათი 1.14. ადმინისტრატორის მხარის routes

როუტები, რომლებიც პასუხისმგებელია სამართავი პანელის ლექციებისა და ლექტორების გვერდებზე.

```

1 Route::prefix('lecturers')->group(function () {
2     Route::get('/', [LecturersAdminController::class, 'List'])->name('admin_lecturers');
3     Route::get('/data.json', [LecturersAdminController::class, 'Json'])->name('admin_lecturers.json');
4     Route::post('/upload', [LecturersAdminController::class, 'upload'])->name('admin_lecturers.upload');
5     Route::get('/run-import/{id}', [LecturersAdminController::class, 'runImport'])->name('admin_lecturers.run_import');
6     Route::post('/post-add', [LecturersAdminController::class, 'Add'])->name('admin_lecturers.post_add');
7     Route::get('/search-lecturers.json', [LecturersAdminController::class, 'searchJson'])->name('admin_lecturers.search_json');
8     Route::get('/load-time-checkboxes', [LecturersAdminController::class, 'TimeCheckBoxes'])->name('admin_lecturers.time_checkboxes');
9     Route::post('/post-edit', [LecturersAdminController::class, 'Edit'])->name('admin_lecturers.post_edit');
10    Route::post('/post-delete', [LecturersAdminController::class, 'Delete'])->name('admin_lecturers.post_delete');
11    Route::post('/post-update-merged-groups', [LecturersAdminController::class, 'UpdateMergedGroups'])->name('admin_lecturers.post_edit_merged_groups');
12    Route::any('/post-schedule-lecture', [LecturersAdminController::class, 'runScheduleLecture'])->name('admin_lecturers.post_schedule_lecture');
13
14 });
15
16 Route::prefix('lectures')->group(function () {
17     Route::get('/overview', [LecturesAdminController::class, 'Overview'])->name('admin_lectures_overview');
18     Route::get('/table/{group_name}', [LecturesAdminController::class, 'List'])->name('admin_lectures_table');
19     Route::get('/data.json', [LecturesAdminController::class, 'Json'])->name('admin_lectures.json');
20     Route::post('/upload', [LecturesAdminController::class, 'upload'])->name('admin_lectures.upload');
21     Route::get('/run-import/{id}', [LecturesAdminController::class, 'runImport'])->name('admin_lectures.run_import');
22     Route::post('/post-add', [LecturesAdminController::class, 'Add'])->name('admin_lectures.post_add');
23     Route::post('/post-edit', [LecturesAdminController::class, 'Edit'])->name('admin_lectures.post_edit');
24     Route::post('/post-delete', [LecturesAdminController::class, 'Delete'])->name('admin_lectures.post_delete');
25     Route::get('/search-lectures.json', [LecturesAdminController::class, 'searchJson'])->name('admin_lectures.search_json');
26     Route::get('/load-table/{id}', [LecturesAdminController::class, 'LoadTable'])->name('admin_lectures_load.table');
27     Route::post('/move-position', [LecturesAdminController::class, 'MovePosition'])->name('admin_lectures.post_move_position');
28
29 });

```

სურათი 1.15. სამართავი პანელის route-ები

ახლა კი წარმოგიდგენთ კოდისა და შესაბამისი ფორმის ფრაგმენტს, რომელიც პასუხისმგებელია, კონკრეტული ჯგუფისათვის, ლექციის ჩასმის ავტომატიზაციაზე. რომელიც ავტომატურად იპოვის თავისუფალ აუდიტორიას და ადრეოულ შესაძლო დროს ლექციის ჩასატარებლად. ასევე იგი ამოწმებს შეუძლია თუ არა ლექტორს ამ დროს ლექციის ჩატარება ამათუიმ დროს (არის თუ არა ლექტორი თავისუფალი), ამოწმებს ლექციის ხანგრძლივობიდან გამომდინარე თავისუფალი აუდიტორიის არსებობას და ასეთების არსებობის შემთხვევაში რანდომულად ჩასვავს აუდიტორიას ლექციისათვის. ასევე შეგვიძლია მივუთითოთ სასურველი კორპუსები, რამოდენიმე ერთად და ჩემს მიერ შექმნილი ალგორითმი სახელად GTUScheduler ავტომატურად იპოვის და აირჩევს მითითებული კორპუსებიდან ერთ რომელიმე თავისუფალ აუდიტორიას. იგი ასევე ამოწმებს დღიურ დატვირთვასაც. ანუ, ამოწმებს დღიური დატვირთვა, ხომ არ სცილდება 8 საათს, რომელიც შრომის კოდექსითაა გათვალისწინებული. და ასევე ამოწმებს კვირეულ დატვირთვა, ხომ არ სცილდება 40 საათს. ასეთ შემთხვევაში კი დაგვიბრუნდება შეტყობინება, რომ ლექციის ჩასმა კონკრეტული ლექტორისათვის ვერ მოხერხდება დღიური თუ კვირეული დატვირთვის ლიმიტის გამო. გარდა ამ ყველაფრის ჩვენ შეგვიძლია ჩავსვათ სხვადასხვა ლექციები სხვადასხვა სემესტრებში. წინასწარ გავამზადოთ მომავალი სემესტრის ცხრილი და ა.შ.

```

1 <?php
2 namespace App\Classes;
3
4 use App\Models\Rooms;
5 use App\Models\Groups;
6 use App\Models\Lectures;
7 use App\Models\BusyHours;
8 use Illuminate\Support\Facades\DB;
9
10 class GTUScheduler
11 {
12     public $schedule_auto = false;
13     public $set_room_auto = false;
14     public $set_inBuildings = [];
15
16     public $title;
17     public $lecturer_id;
18     public $group_id;
19     public $day_id;
20     public $start_at;
21     public $duration;
22     public $ends_at;
23     public $building_code;
24     public $room_code = null;
25     public $semester_id = 0;
26
27     public function setSemester($id){
28         $this->semester_id = intval($id) > 0 ? intval($id) : curentSemester();
29     }

```

სურათი 1.16. GTUScheduler

GTUScheduler კლასის გამოყენების ნუმუში ლექციების კონტროლერში

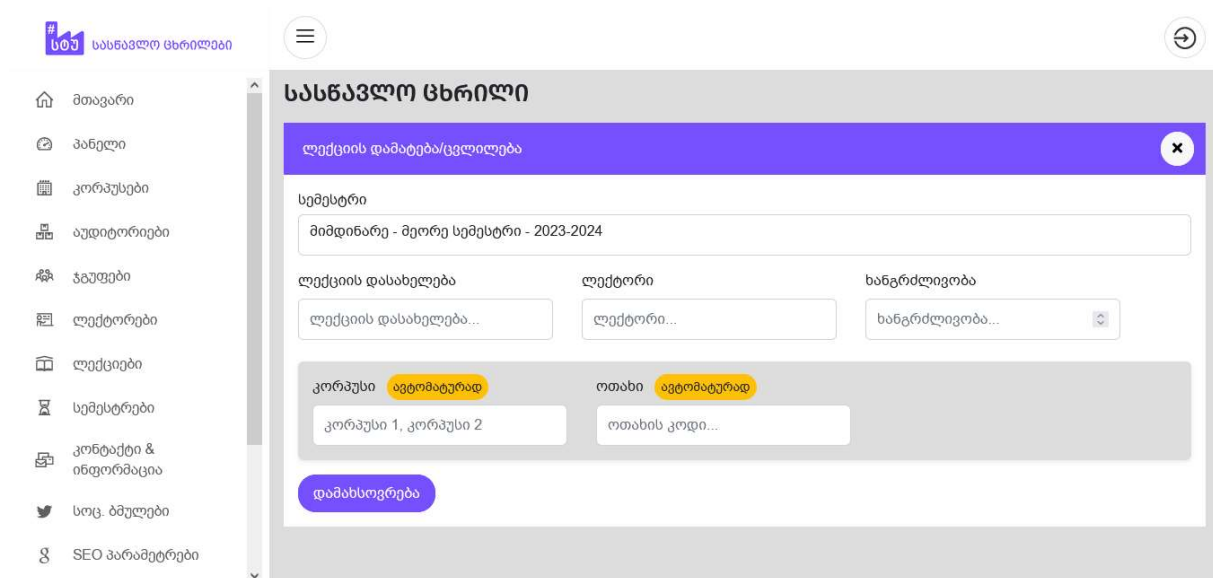

```

1 public function Add(Request $request)
2 {
3     $validation = Validator::make($request->all(), [
4         'title' => ['required', 'max:535'],
5         'duration' => ['required', 'numeric'],
6         'group_id' => ['required', 'numeric', new GroupExists],
7     ]);
8
9     if ($validation->fails()) {
10         return response()->json([
11             'message' => $validation->errors()->first(),
12             'status' => 0,
13             'class' => 'error'
14         ]);
15     }
16
17     // $record_id = intval($request->post('set_building'));
18     $mode = trim($request->post('mode'));
19
20
21     $set_building = array_filter(explode(',', $request->post('set_building')));
22     $room = $request->post('set_room') ?? 'auto';
23     $semester_id = isset($_POST['semester_id']) ? intval($_POST['semester_id']) : 0;
24
25     // set semester id
26     // semester id
27     $run = new GTUScheduler;
28     $run->setSemester($semester_id);
29
30     if (isset($set_building[0]) && $room == 'auto') {
31         $run->setRoomAuto($set_building);
32     } elseif (isset($set_building[0]) && $room == 'auto') {
33         // randomises everything
34         $buildingIds = Buildings::has('rooms')->inRandomOrder()->pluck('id')->toArray();
35         $run->setRoomAuto($buildingIds);
36     } else {
37         $run->setRoom($set_building[0], $room);
38     }
39
40     $lecturer_id = $request->post('lecturer_id');
41     $lecturer_id = $lecturer_id ? intval($lecturer_id) : findOrCreateLecturer($request->post('lecturer_name'));
42     $response = $run->addLecture($request->post('title'), $lecturer_id, $request->post('group_id'), $request->post('duration'));
43
44     return response()->json([
45         'message' => $response['message'],
46         'status' => $response['status'],
47         'class' => $response['status'] == 1 ? 'success' : 'error'
48     ]);
49 }

```

სურათი 1.17. ლექციების კონტროლერი

ასე გამოიყურება ჯგუფისთვის ლექციის ჩასმის გვერდი, რომელიც დაკავშირებულია ჩემს მიერ უკვე აღნიშნულ ლოგიკასთან



სურათი 1.18. ლექტორისთვის საგნის ჩამატება

გრადა ზემოთ აღნიშნულისა, ასევე შევქმენით GTUMultiScheduler კლასი, რომელსაც გააჩია თითქმის იგივე დანიშნულება ოღონდ, იგი განკუთვნილია ლექტორის გვერდისათვის, სადაც გვაქვს საშუალება ერთდროულად ჩავუსვათ საგანი რამდენიმე ჯგუფს, დანარჩენი კი თითქმის იგივეა.

```
1 public function runScheduleLecture(Request $request)
2 {
3     $validation = Validator::make($request->all(), [
4         'subject' => ['required', 'max:535'],
5         'lecturer_id' => ['required', 'numeric', new LecturerExists],
6         'setduration' => ['required', 'numeric'],
7         //'group_id' => ['required', 'numeric', new GroupExists],
8     ]);
9
10    if ($validation->fails()) {
11        return response()->json([
12            'message' => $validation->errors()->first(),
13            'status' => 0,
14            'class' => 'error'
15        ]);
16    }
17
18    $set_building = array_filter(explode(',', $request->post('set_building')));
19    $room = $request->post('set_room') ?? 'auto';
20    $semester_id = isset($_POST['semester_id']) ? intval($_POST['semester_id']) : 0;
21
22    // set semester id
23    // semester_id
24    $run = new GTUMultiScheduler;
25    $run->setSemester($semester_id);
26
27    if (isset($set_building[0]) && $room == 'auto') {
28        $run->setRoomAuto($set_building);
29    } elseif (!isset($set_building[0]) && $room == 'auto') {
30        // randomies everything
31        $buildingIds = Buildings::has('rooms')->inRandomOrder()->pluck('id')->toArray();
32        $run->setRoomAuto($buildingIds);
33    } else {
34        $run->setRoom($set_building[0], $room);
35    }
36
37    $set_groups = $request->post('set_groups');
38    $set_groups = array_filter(explode(',', $set_groups));
39    $lecturer_id = $request->post('lecturer_id');
40    $lecturer_id = $lecturer_id ? intval($lecturer_id) : findOrCreateLecturer($request->post('lecturer_name'));
41    $response = $run->addLecture(
42        $request->post('subject'),
43        $lecturer_id,
44        $set_groups,
45        $request->post('setduration')
46    );
47
48    return response()->json([
49        'message' => $response['message'],
50        'status' => $response['status'],
51        'class' => $response['status'] == 1 ? 'success' : 'error'
52    ]);
53 }
```

სურათი 1.19. GTUMultiScheduler - გამოყენების ფრაგმენტი

ქვემოთ მოცემული კოდი საშუალებას იძლევა მივამაგროთ სასწავლო ჯგუფები ლექტორთან. ან პირიქით, ჩავხსნათ ლექტორს რომელიმე მიმდგრებული ჯგუფი.

```

1 public function UpdateMergedGroups(Request $request)
2 {
3     $lecturer_id = intval($request->post('lecturer_id'));
4     $semester_id = intval($request->post('semester_id'));
5     $selected = trim($request->post('selected'));
6     $selected = array_filter(explode(',', $selected));
7     $selected = implode(',', $selected);
8
9     $check = Lecturers::where('id', $lecturer_id);
10    if ($check->count() == 0) {
11        return response()->json([
12            'message' => 'ლექტორი ვერ მოიძებნა',
13            'status' => 0,
14            'class' => 'error'
15        ]);
16    }
17
18    $semester = Semesters::where('id', intval($semester_id));
19    if ($semester->count() == 0) {
20        return response()->json([
21            'message' => 'სემესტრი ვერ მოიძებნა',
22            'status' => 0,
23            'class' => 'error'
24        ]);
25    }
26
27    $mergeds = MergedGroups::where('semester_id', intval($semester_id))
28        ->where('lecturer_id', intval($lecturer_id));
29
30    if ($mergeds->count()) {
31        // edit
32        MergedGroups::where('id', $mergeds->first()->id)->update(['groups' => $selected]);
33    } else {
34        MergedGroups::insert(['semester_id' => $semester_id, 'lecturer_id' => $lecturer_id, 'groups' => $selected, 'created_at' => now(), 'updated_at' => now()]);
35    }
36
37    return response()->json([
38        'message' => 'მიმაგრებული ჯგუფების სია წარმატებით განახლდა',
39        'status' => 1,
40        'class' => 'success'
41    ]);
42 }

```

სურათი 1.20. ჯგუფების მიმაგრება

აქ კი წარმოდგენილია კოდის ფრაგმენტი რომელიც უზრუნველყოფს ლექტორებისათვის „დაკავებული საათები“-ს ჩამატებას და ამოკლებას. საათებისა და დღეების დაკავებულ/არათავისუფალ დროს მონიშვნას.

```

1 public function Edit(Request $request)
2 {
3     $validation = Validator::make($request->all(), [
4         'title' => 'required|max:200'
5     ]);
6
7     if ($validation->fails()) {
8         return response()->json([
9             'message' => $validation->errors()->first(),
10            'status' => 0,
11            'class' => 'error'
12        ]);
13    }
14
15    $check = Lecturers::where('id', intval($request->post('record_id')));
16    if ($check->count() == 0) {
17        return response()->json([
18            'message' => 'ჩანაწერი ვერ მოიძებნა',
19            'status' => 0,
20            'class' => 'error'
21        ]);
22    }
23
24    $lecturer_name = trim($request->post('title'));
25    $id = $check->first()->id;
26
27    Lecturers::where('id', $check->first()->id)->update([
28        'name_ge' => latin_to_georgian($lecturer_name),
29        'name_en' => georgian_to_latin($lecturer_name),
30    ]);
31
32    // clear busy hours
33    BusyHours::where('lecturer_id', $id)->delete();
34
35    $days = [
36        trim($request->post('day_1')),
37        trim($request->post('day_2')),
38        trim($request->post('day_3')),
39        trim($request->post('day_4')),
40        trim($request->post('day_5')),
41        trim($request->post('day_6'))
42    ];
43
44    foreach ($days as $i => $busyHourInput) {
45        $i++;
46        BusyHours::insert([
47            'day_id' => $i,
48            'lecturer_id' => $id,
49            'hours' => trim($busyHourInput),
50            'created_at' => now(),
51            'updated_at' => now()
52        ]);
53    }
54
55    return response()->json([
56        'message' => 'ჩანაწერი წარმატებით განახლდა',
57        'status' => 1,
58        'class' => 'success'
59    ]);
60 }

```

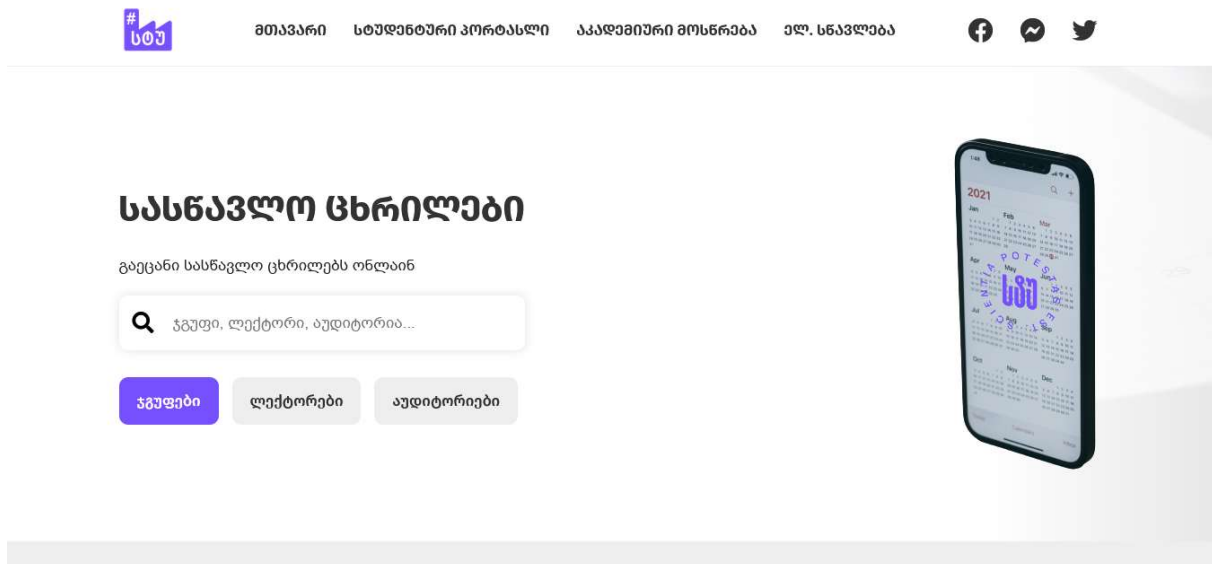
სურათი 1.21. BusyHours რედაქტირება

1.3. ვებგვერდის ფუნქციონალის დემონსტრირება

მთავარ გვერდის ძირითად ნაწილში განთავსებულია ფსევდო/დეკორატიული საძიებო ფორმა, რომელზე დაკლიკების შემდეგაც იხსნება მოდალური პოპაფ ფანჯარა და კურსორი ავტომატურად გადადის რეალურ საძიებო ველზე, რომელიც განთავსებულია აღნიშნულ მოდალურ ფანჯარაზე. აღნიშნულ საძიებო ველში მომხმარებელს შეუძლია მოიძებნოს სასწავლო ჯგუფების, პედაგოგების და აუდიტორიული დატვირთვის ჯგუფები და ამისათვის სულ რამდენიმე წამია საჭირო.

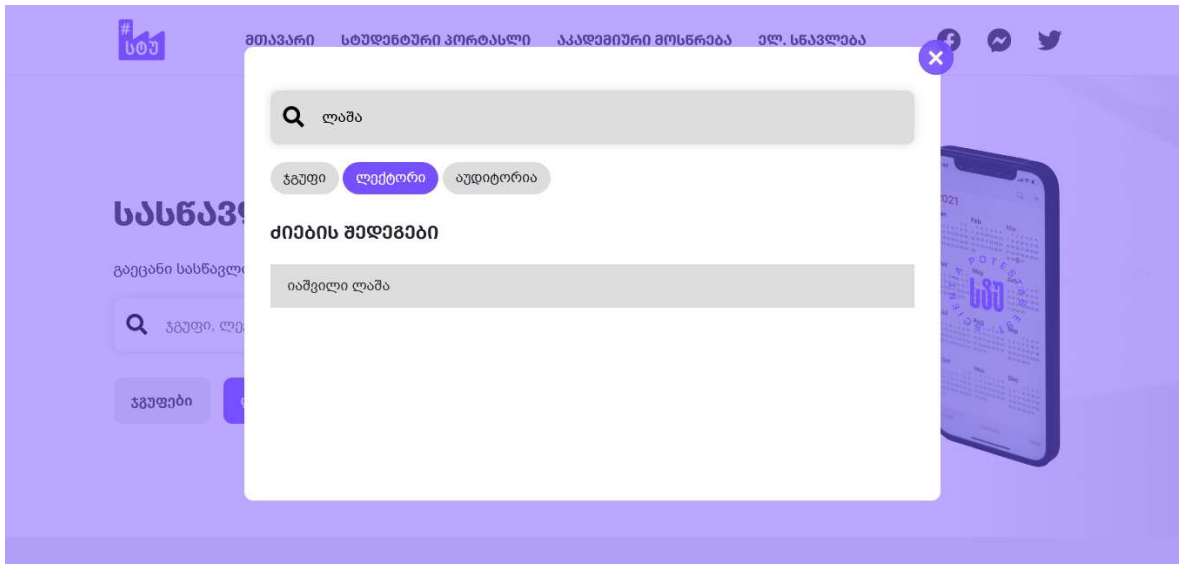
გარდა ამისა, მთავარ გვერდზე წარმოდგენილია უნივერსიტეტის სხვადასხვა რესურსებთან წვდომის ბმულები, რათა სტუდენტებმა მომენტალურად შეძლონ მათთვის მნიშვნელოვან საიტებსა და სასწავლო რესურსებზე წვდომა.

ვებსაიტის მთავარი გვერდი, მთავარი ნავიგაციური მენიუ, გვერდის ძირითადი ნაწილი ფსევდო საძიებო ფორმითა და ძიების ფილტრებით:



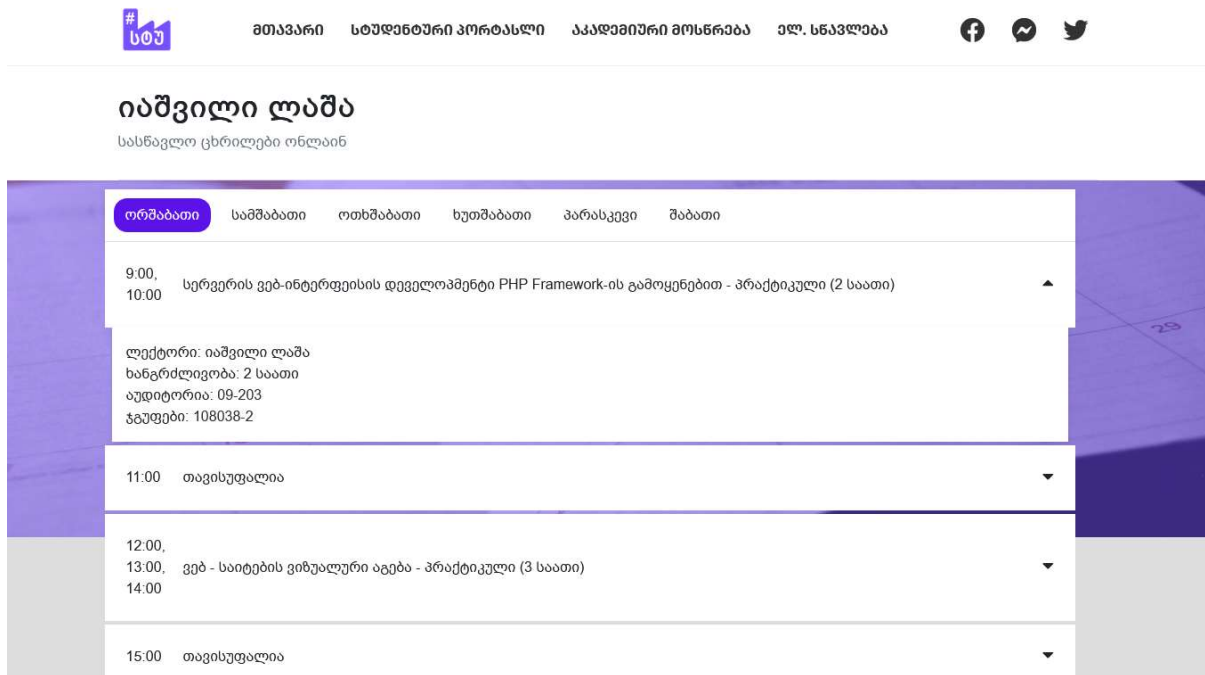
სურათი 1.22. საიტის მთავარი სექცია

რეალური ძიების ფორმა და სპეციალურად ცხრილების ძიებისთვის შექმნილი მოდალური პოპაფ ფანჯარა. ფოტოზე ასევე თქვენ ხედავთ ლექტორის ძიების შედეგს, რომელზე დაკლიკების შემდეგაც გადავდივართ კონკრეტული ლექტორის ცხრილებზე.



სურათი 1.23. ლექტორის ძებნა

ლექტორის ცხრილი, ლექციების ჩატარების განრიგი



სურათი 1.24. ლექტორის ცხრილი

ჯგუფი: 108938-1

სასწავლო ცხრილები ელექტრონულ ფორმატში

ორშაბათი	სამშაბათი	ოთხშაბათი	ხუთშაბათი	პარასკევი	შაბათი
9:00, 10:00	კომპიუტერული ბუღალტრული აღრიცხვა - პრაქტიკული (2 საათი)				
11:00	თავისუფალია				
12:00	საბაკალავრო პროექტი (INF) - საკურსო (1 საათი)				
13:00, 14:00	ინფორმაციული სისტემების პროექტების მართვის საფუძვლები - პრაქტიკული (2 საათი)				
15:00	ინფორმაციული სისტემების პროექტების მართვის საფუძვლები - ლექცია (1 საათი)				
16:00	IT მენარმეობა და სტანდარტები - ლექცია (1 საათი)				

სურათი 1.25. სასწავლო ჯგუფის ცხრილი

აუდიტორია: 06-405ბ

აუდიტორიის სასწავლო დატვირთვის ცხრილი

ორშაბათი	სამშაბათი	ოთხშაბათი	ხუთშაბათი	პარასკევი	შაბათი
9:00	თავისუფალია				
10:00	თავისუფალია				
11:00	თავისუფალია				
12:00	საბაკალავრო პროექტი (INF) - საკურსო (1 საათი)				
ლექტორი: კოტრიკაძე გულნარა ხანგრძლივობა: 1 საათი აუდიტორია: 06-405ბ ჯგუფები: 108938-1					
13:00	თავისუფალია				

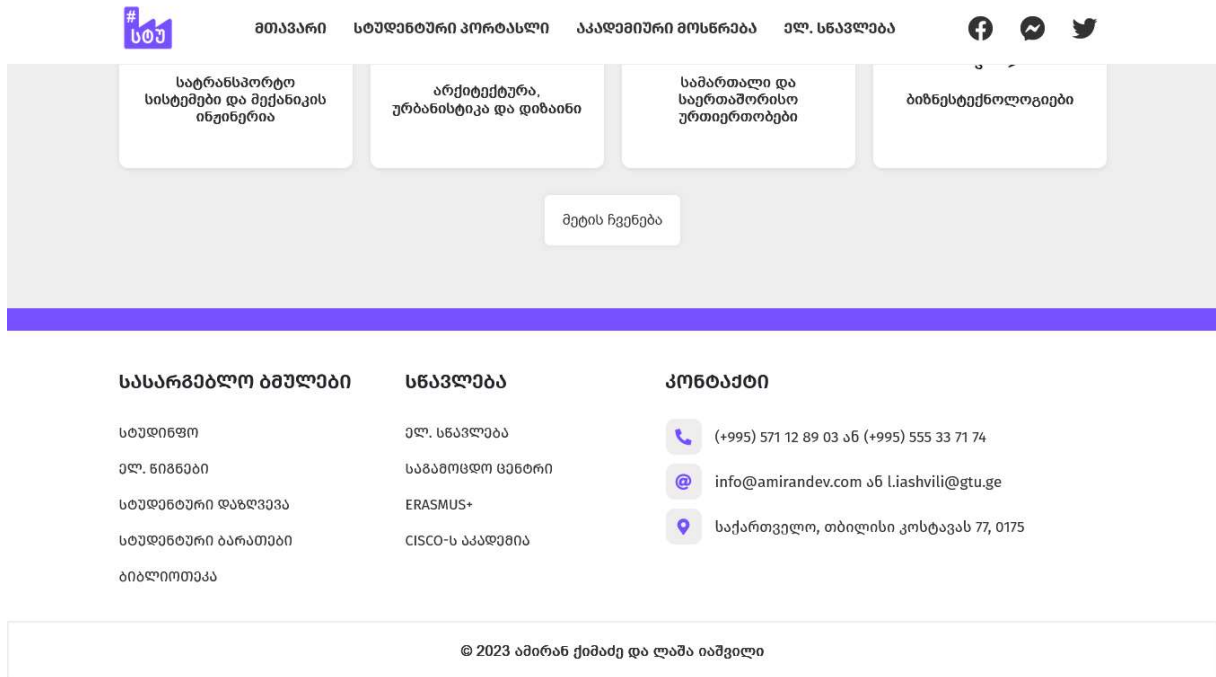
სურათი 1.26 აუდიტორიის სასწავლო დატვირთვის ცხრილი

საქართველოს ტექნიკური უნივერსიტეტის ფაკულტეტების ჩამონათვალი ბმულებით, შესაბამის რესურსებზე სწრაფი წვდომისათვის:

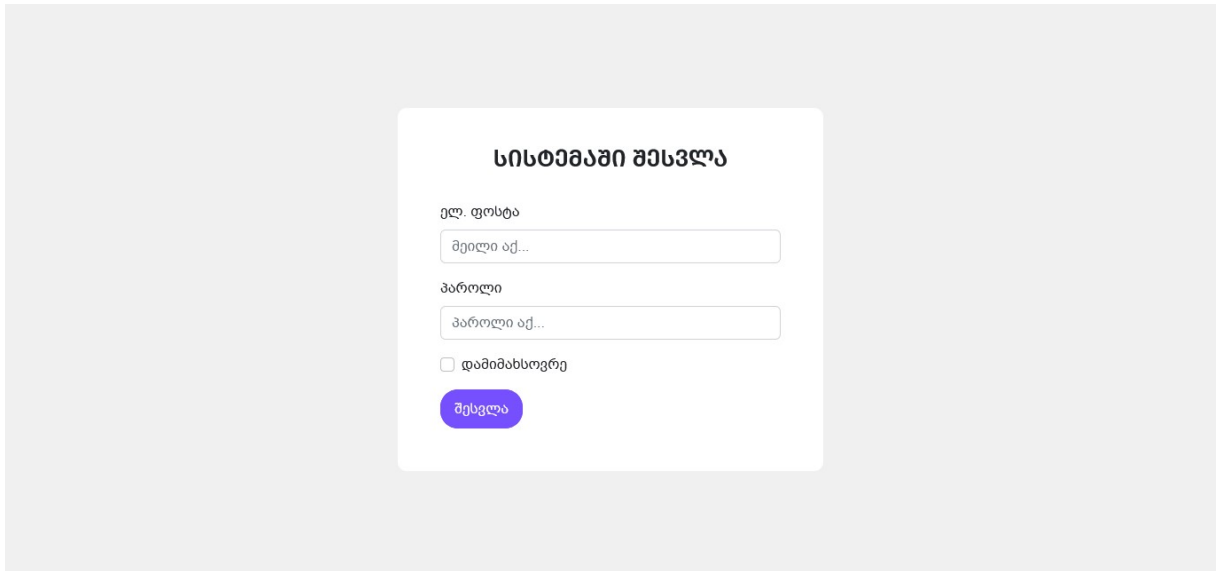
ფაკულტეტები



სურათი 1.27. ფაკულტეტების ჩამონათვალი

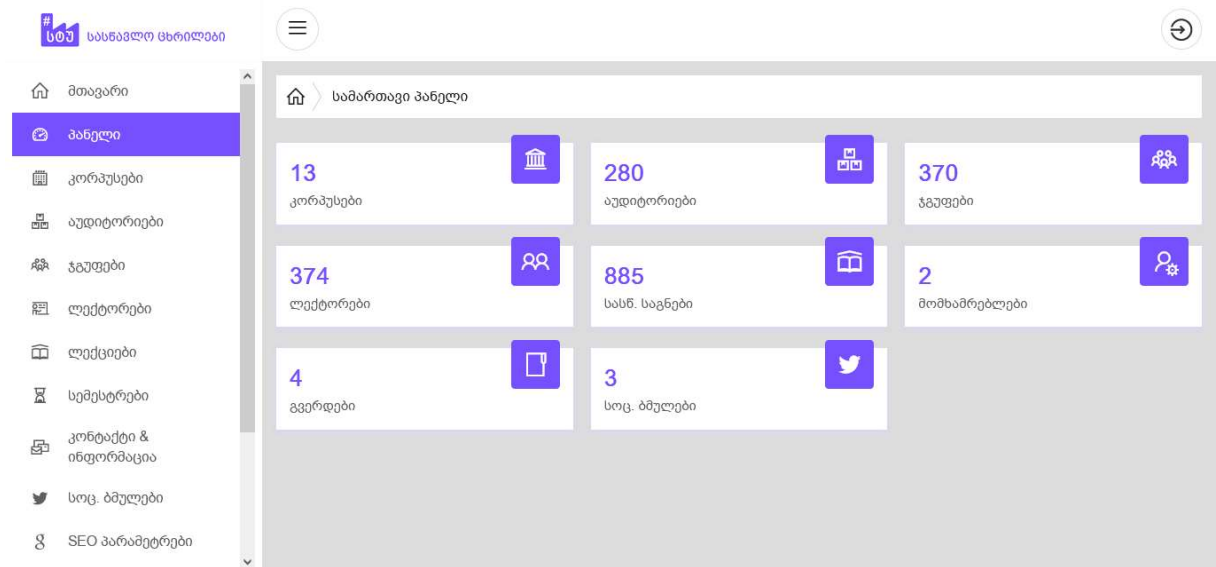


სურათი 1.28. საკონტაქტო ინფორმაციის ბლოკი საიტზე



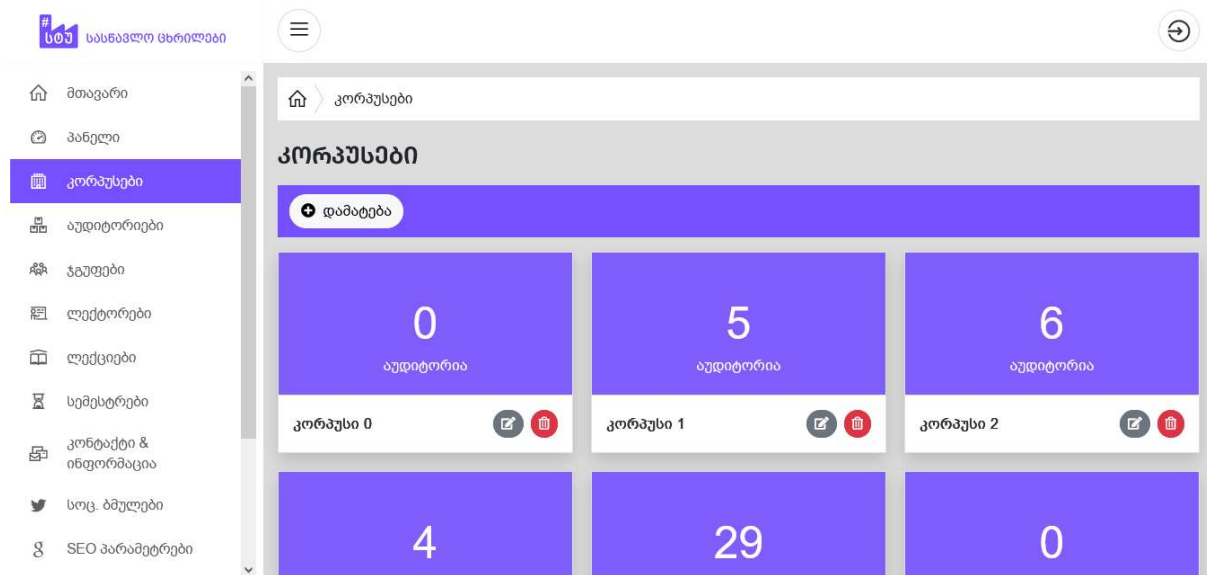
სურათი 1.29. სამართავ პანელში შესვლის გვერდი

ვებსაიტის სამართავი პანელის ძირითადი გვერდი. სადაც წარმოდგენილია ზოგადი სტატისტიკა მონაცემთა ბაზაში არსებული ჩანაწერების შესახებ. როგორცაა: კორპუსები, აუდიტორიები, ჯგუფები, ლექტორები, სასწავლო საგნები და სხვა.



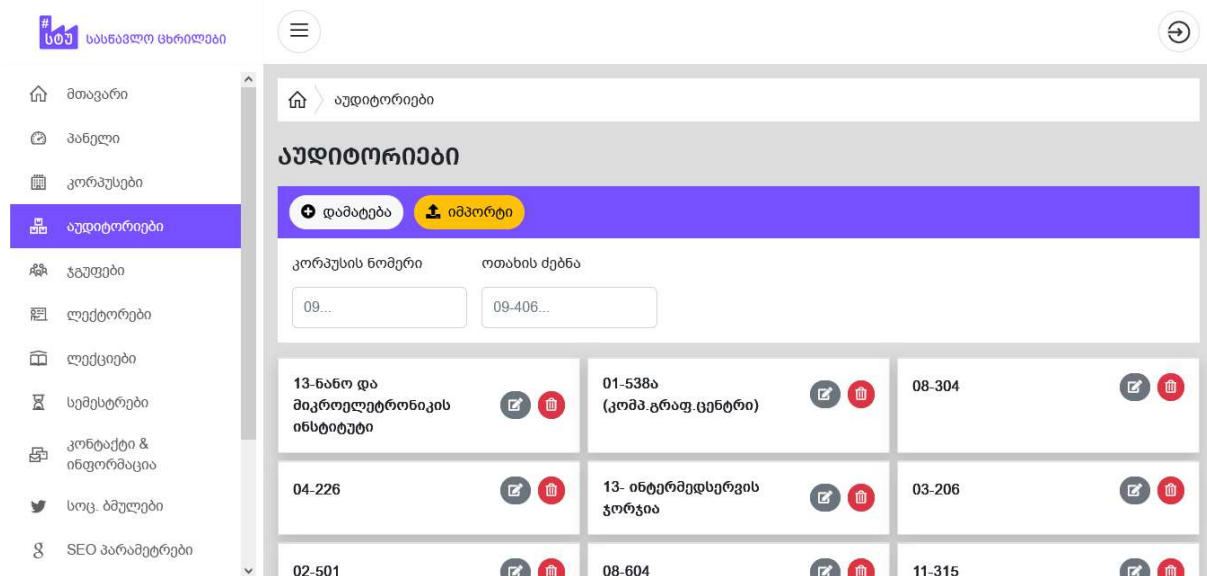
სურათი 1.30. სამართავი პანელის მთავარი გვერდი

სამართავი პანელის კორპუსების გვერდიდან შესაძლებელია კორპუსების დამატება, რედაქტირება წაშლა, კორპუსებში აუდიტორიების რაოდენობის ნახვა და სხვა



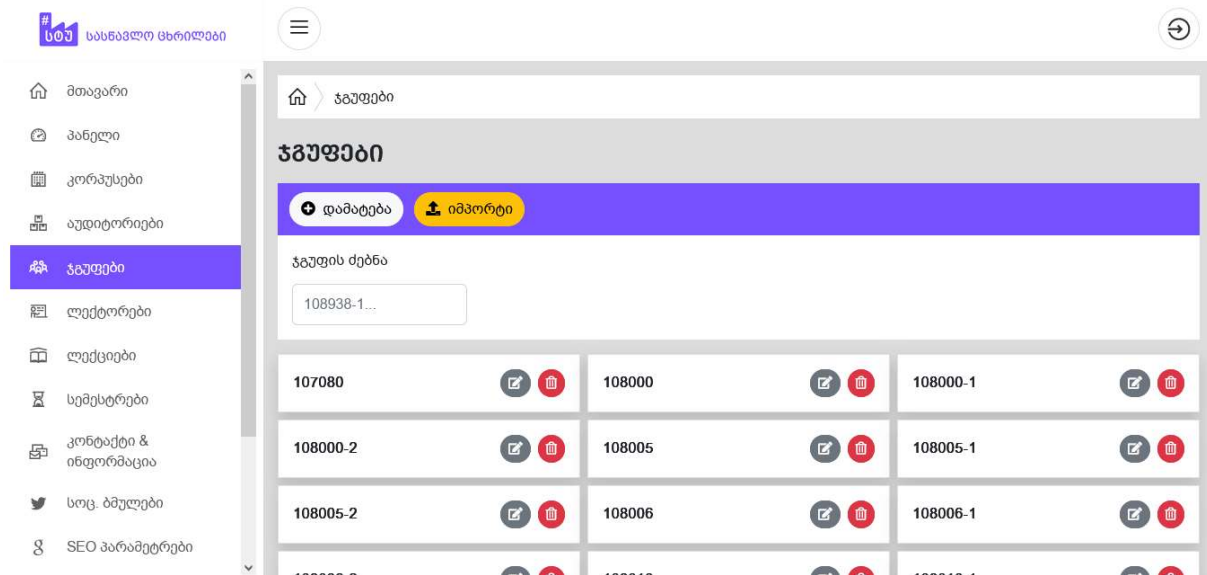
სურათი 1.31. კორპუსების გვერდი

სამართავი პანელის აუდიტორიების გვერდიდან შესაძლებელია აუდიტორიების ძებნა, დამატება, რედაქტირება, აშლა და excel-იდან აუდიტორიების სიის იმპორტირება:



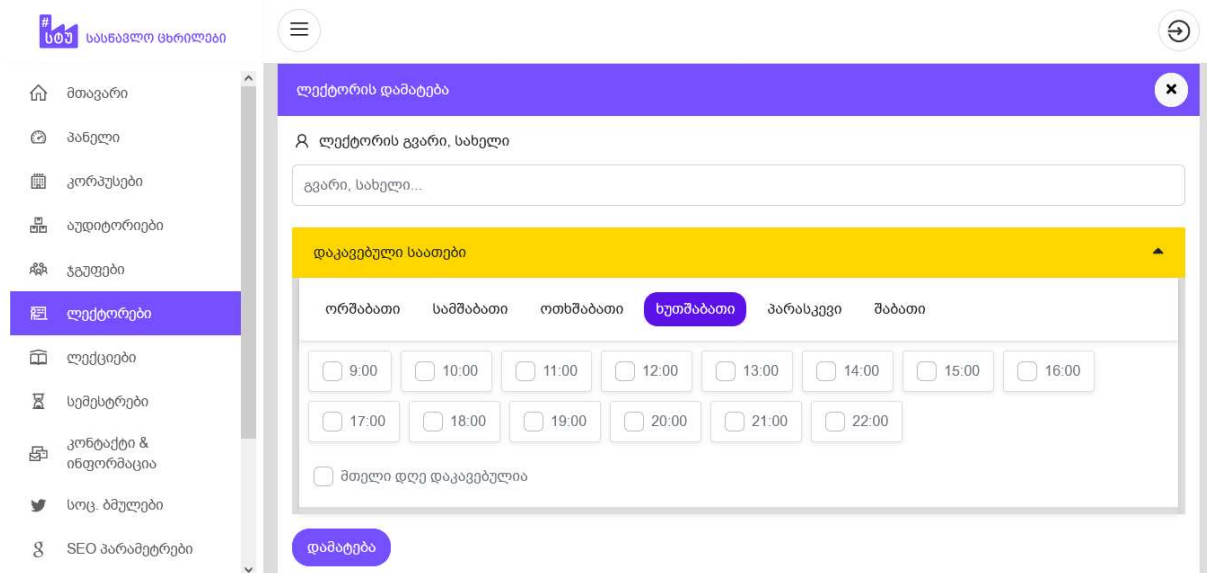
სურათი 1.32. აუდიტორიების გვერდი

სამართავი პანელის სასწავლო ჯგუფების გვერდიდან შესაძლებელია სასწავლო ჯგუფების დამატება, რედაქტირება, წაშლა, სასწავლო ჯგუფების შესაბამის სასწავლო ცხრილებზე გადასვლა

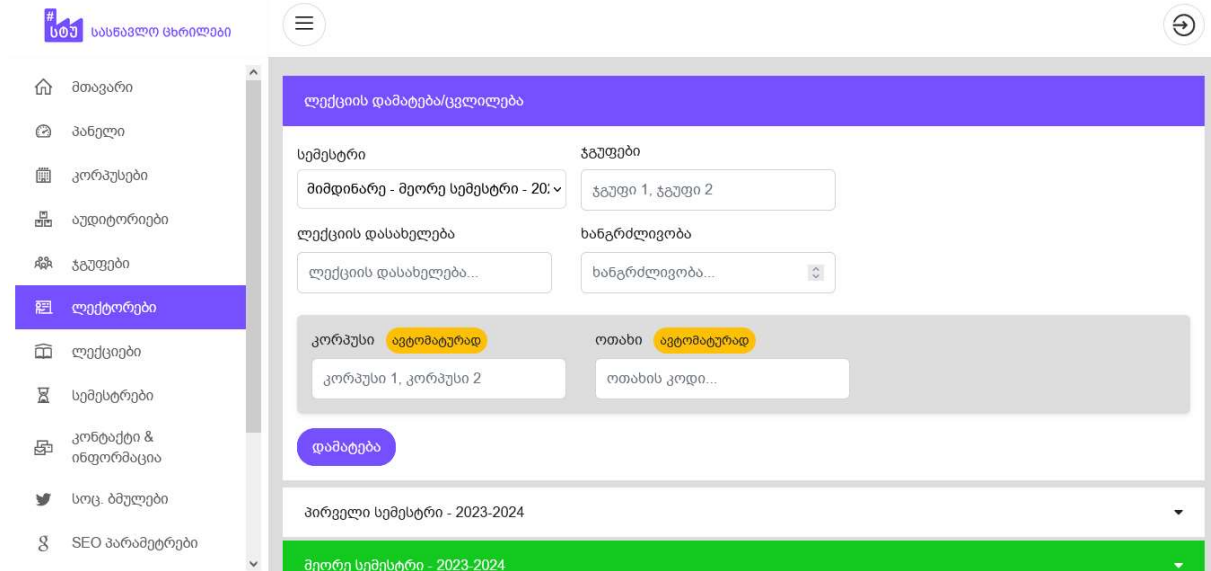


სურათი 1.33. ჯგუფების გვერდი

სამართავი პანელის ლექტორების გვერდიდან შესაძლებელია ლექტორების დამატება, რედაქტირება, წაშლა, ლექტორისათვის საათებისა და დღეების დაკავებულ საათებათ და დაკავებულ/არათავისუფალ დღეებად მონიშვნა. ლექტორისთვის ჯგუფების მიმაგრება. სასწავლო ჯგუფებისთვის ამათუ-იმ ლექტორის ლექციის ჩასმა სასურველი სემესტრის სასწავლო ცხრილში.

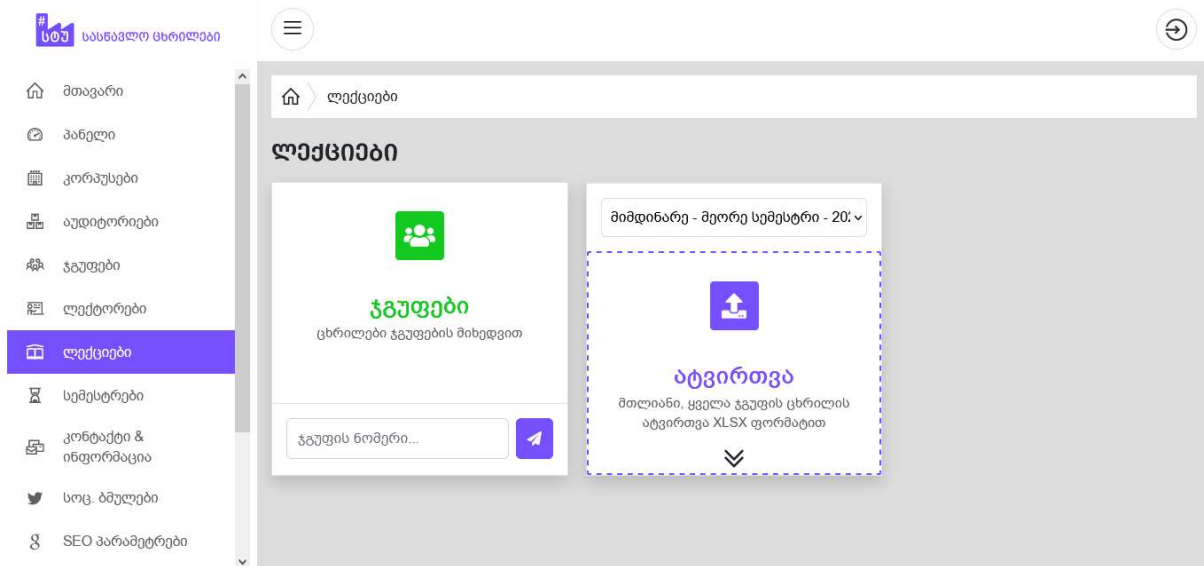


სურათი 1.34. დაკავებული საათების მონიშვნა



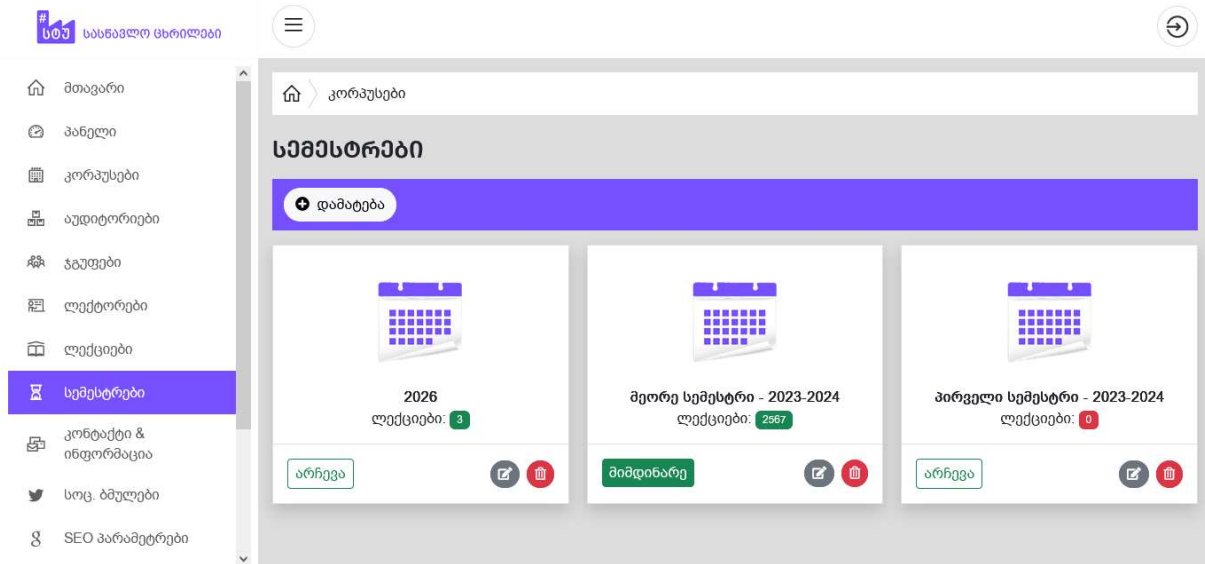
სურათი 1.35. ლექტორისთვის საგნის ჩასმა.

სამართავი პანელის ლექციების გვერდიდან შესაძლებელია ნავიგაცია ჯგუფების სასწავლო ცხრილებზე და და სასწავლო ცხრილების იმპორტი Excel-ის ფაილიდან ამა-თუ-იმ სემესტრისათვის



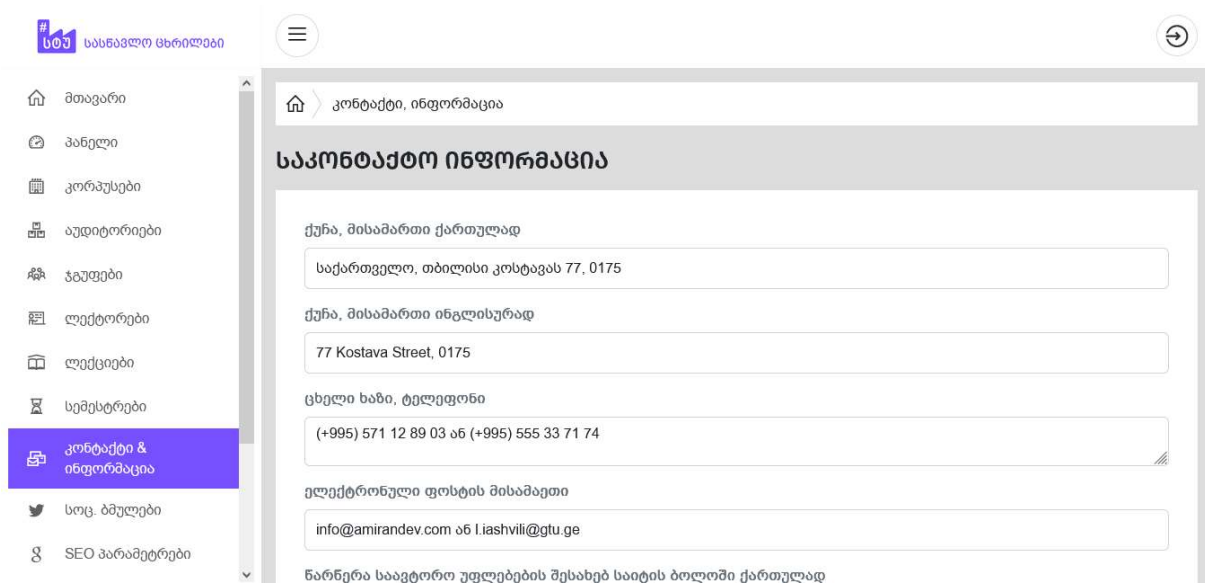
სურათი 1.36. ლექციების გვერდი

სამართავი პანელის სემესტრების გვერდიდან შესაძლებელია სემესტრების დამატება, რედაქტირება, სემესტრისა და სემესტრის ცხრილების წაშლა. ჩანიშნული ლექციების ოდენობის ნახვა თითოეული სემესტრისათვის და სემესტრის მიმდინარედ მონიშვნა.



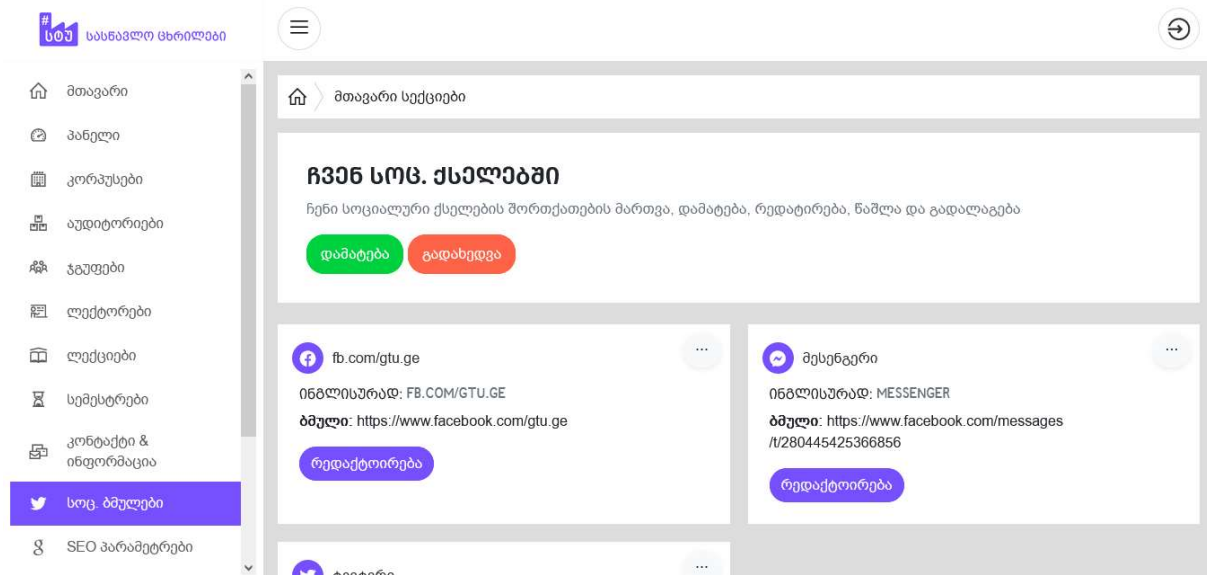
სურათი 1.37. სემესტრების გვერდი

სამართავი პანელის „კონტაქტი და ინფორმაცია“ გვერდიდან შესაძლებელია საიტზე განთავსებული საკონტაქტო ინფორმაციის განახლება



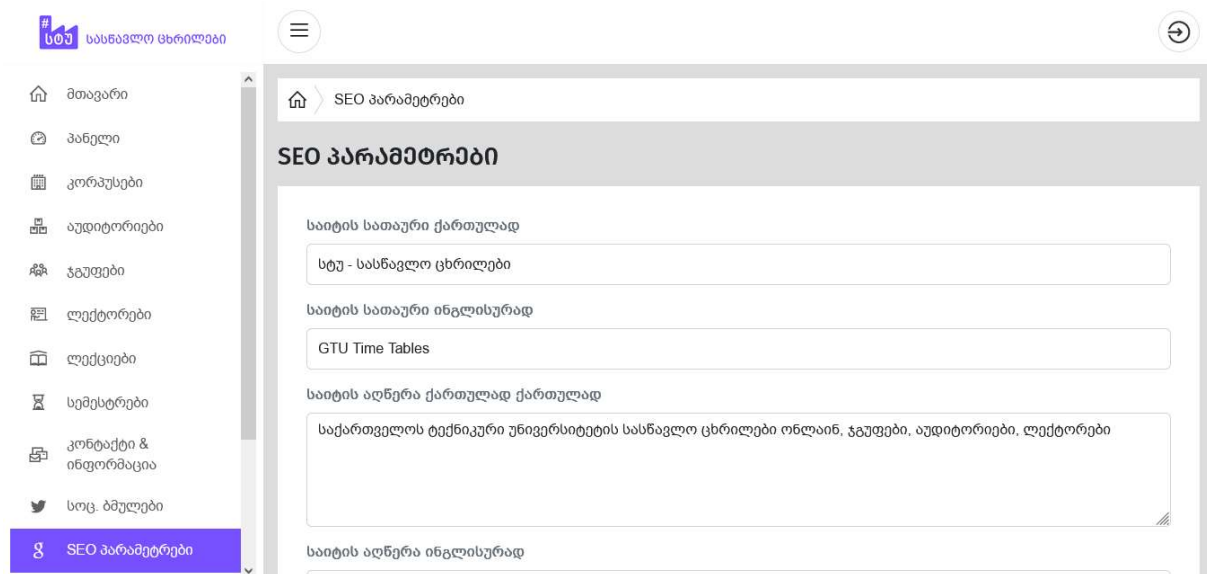
სურათი 1.38. საკონტაქტო ინფორმაციის გვერდი

სამართავი პანელის „სოც. ბმულები“ გვერდიდან შესაძლებელია საიტის საჯარო (public) გვერდებზე, მთავარ ნავიგაციურ მენიუზე განთავსებული საქართველოს ტექნიკური უნივერსიტეტის სოციალური მედიის გვერდების ბმულების დამატება, რედაქტირება, გადანაცვლება და წაშლა.



სურათი 1.39. სოც. ბმულების გვერდი

სამართავი პანელის „SEO პარამეტრები“ გვერდიდან შესაძლებელია საიტის meta ინფორმაციის, **fb:app_id** და Open graph image ცვლილება.



სურათი 1.40. SEO პარამეტრები

სამართავი პანელის პარამეტრების გვერდზე მოცემულია საიტის ანიმაციური “Loader”-ის ჩართვა/გამორთვის ღილაკი და საიტის ენის არჩევის “select input”. თუმცა ამ მომენტისთვის ხელმისაწვდომია მხოლოდ ქართული ენა.

პროექტის დასრულებამდეც ვიყენებდით შექმნილ პლატფორმას ყოველდღიურად, ნებისმიერი ჯგუფის სასწავლო ცხრილის, სალექციო განრიგისა და თავისუფალ დროს, ამავე პროექტზე სამუშაოთ საქართველოს ტექნიკურ უნივერსიტეტში თავისუფალი აუდიტორიის საპოვნელად.

აღნიშნული პროექტი - **ონლაინ სასწავლო ცხრილების პლატფორმა GTU TABLES**, გახდა საფუძველი იმისა, რომ მომხდარიყო საქართველოს ტექნიკური უნივერსიტეტის ცხრილების განახლება, ახალი სტანდარტებითა და შესაბამისად ყველანაირი ზემოაღნიშნული კუთხით, მომხდარიყო ცხრილებთან გამარტივებული და გაუმჯობესებული წვდომა, ყველა მიმართულებით.

მიღებული პროექტი, დადებითად იქნა შეფასებული საქართველოს ტექნიკური უნივერსიტეტის, ინფორმატიკისა და მართვის სისტემების ფაკულტეტის მიერ და ამჟამად მიმდინარეობს მისი დანერგვა და შესაბამისი ლავირება აღნიშნული პროექტის, რომელიც იქნება მაქსიმალურად მორგებული შიდა სასწავლო პროცესზე.

ბიბლიოგრაფია

1. ლაშა იაშვილი
HTML 5-ის შესაძლებლობები, გვ.314. თბილისი 2017წ.
https://gtu.ge/book/L_iashvili_HTML.pdf
2. David Sklar და Adam Trachtenberg
PHP Cookbook (მესამე გამოცემა) O'Reilly Media, Inc. 2014
<https://freepdf-books.com/php-cookbook-3rd-edition-pdf-books/>
3. PHP Foundation & Community
ოფიციალური დოკუმენტაცია 2023 (ონლაინ რესურსი)
<https://www.php.net/docs.php>
4. Taylor Otwell & Community
ლარაველის ოფიციალური დოკუმენტაცია (ონლაინ რესურსი) 2023წ
<https://laravel.com/docs>
5. Timmy Willison
Jquery - ოფიციალური დოკუმენტაცია (ონლაინ რესურსი) 2023წ
<https://api.jquery.com/>
6. Mark Otto & Jacob Thornton & Community
Bootstrap 5.x ოფიციალური დოკუმენტაცია (ონლაინ რესურსი) 2023წ
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

Online Learning Tables Platform GTU TABLES

Amiran Qimadze¹; Gulnara Kotrikadze²

¹Georgian Technical University, Faculty of Informatics and Management Systems, Bachelor's degree graduate, Chief specialist; ²Georgian Technical University, Faculty of Informatics and Management Systems, Associate Professor

Resume

We have always been worried about the inconvenience of the current online table platform of the Technical University, leqtori.gtu.ge, because it is quite heavy and it is difficult to find study tables and view them in time. In addition, it is still difficult to use. Its complexity depends on the device, mobile, computer, etc. on technical characteristics, RAM capacity and even internet speed. One of the main reasons for all this is that on the relevant pages of the current educational tables platform, the tables of all study groups, audiences and lecturers of the university are loaded together, which makes it very difficult to find the desired table. In order to clarify all this, I will tell you that when using the online tables of Georgia Technical University on a mobile platform with a smartphone that has 2GB of RAM and we use a 4G Internet connection, the average time required to find and open the study table of the desired group is from 1.7 to 2 minutes. From which the time required to load the page is about 1.6 seconds. Which is really not good. Moreover, if we look at the statistics provided by Google, the technology giant and the world's number one search engine today, we will read that by 2023, the average page load time for a computer will be 2.5 - 8.6 seconds, and the average mobile website load time on a 3G connection will be 19 seconds.

We decided to create a new system of online study tables for the Technical University of Georgia, which will improve the university life of every teacher and student of the university, and at the same time, it will further raise the prestige of our respected university, which educates many professionals and has a very long history.

Based on these and other motives, we have created and present to you the project called "GTU TABLES - educational tables platform".