

ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი

გიორგი კახურაშვილი

## CS - ვებ კონტენტის მართვის სისტემა (CMS)

სამაგისტრო პროგრამა - ინფორმაციული ტექნოლოგიები

ნაშრომი შესრულებულია ინფორმაციული ტექნოლოგიების მაგისტრის აკადემიური  
ხარისხის მოსაპოვებლად

ხელმძღვანელი:

ასოცირებული პროფესორი მაგდა ცინცაძე

თბილისი

2019

# სარჩევი

ანოტაცია.....	4
Abstract .....	5
შესავალი.....	6
CMS - კონტენტის მართვის სისტემა.....	6
სხვადასხვა ტიპის ინფორმაციის მართვის სისტემები.....	6
საჯარო CMS:.....	7
მესაკუთრეობრივი .....	8
პროგრამული უზრუნველყოფა, როგორც სერვისი (SAAS) CMS .....	9
MYSQL.....	10
ვებ ტექნოლოგიები.....	10
Ruby .....	11
Java .....	11
.NET Framework .....	12
Python .....	12
PHP .....	13
პოპულარული PHP ფრეიმვორკები.....	14
Laravel .....	14
CodeIgneter .....	15
Symfony.....	16
MVC (model-view-controller) .....	16
კომპონენტები.....	17
დასმული ამოცანის გადასაჭრელად დაგეგმილი ეტაპები:.....	18
ვებ გვერდის ტექნიკური დავალება:.....	20
ამოცანის პრაქტიკული რეალიზაცია .....	27

საიტის ინტერფეისის აწყობა.....	27
მთავარი გვერდი .....	27
.....	28
სიახლეების ჩამონათვალი .....	29
აკადემიური პერსონალის გვერდი.....	30
ვაკანსიების გვერდი.....	31
გაცვლითი პროგრამების გვერდი .....	32
ტექსტური გვერდები.....	33
კონტაქტის გვერდი.....	34
რეგისტრაციის ფორმის გვერდი .....	35
საიტის სამართავი პანელის ინტერფეისის აწყობა.....	36
მთავარი გვერდი .....	36
ჩამონათვალის ინტერფეისი .....	37
დამატების ინტერფეისი .....	38
რედაქტირების ინტერფეისი.....	39
ძრავი.....	41
ლარაველის ინსტალაცია .....	41
ადმინისტრატორის ავტორიზაცია.....	41
საიტის სტრუქტურის შესაბამისი ცხრილების შექმნა .....	42
მართვის პანელის ინტერფეისის ძრავთან დაკავშირება .....	45
ვებ-გვერდის სერვერზე გამართვა .....	54
დასკვნა.....	55
ლიტერატურა.....	57

## ანოტაცია

სამაგისტრო ნაშრომის მიზანი იყო შექმნილიყო პროგრამული უზრუნველყოფის პროდუქტი (CMS), რომელიც კომპიუტერულ მეცნიერებათა დეპარტამენტის რესურსების შესახებ ინფორმაციის საჯაროდ ხელმისაწვდომობას უზრუნველყოფდა, კერძოდ შექმნილიყო ვებ-გვერდი და მისი მართვის სისტემა, რომელიც კონკრეტულად კომპიუტერული მეცნიერების პროგრამას, სტუდენტებს, ლექტორებსა და დეპარტამენტის საქმიანობას ასახავდა და მათ მოემსახურებოდა. მართალია თბილისის სახელმწიფო უნივერსიტეტს აქვს ვებ-გვერდი, მაგრამ მასზე ძალზედ მწირი ინფორმაციაა განთავსებული კონკრეტულად ჩვენი დეპარტამენტის შესახებ და ამასთან გვერდის განახლება, ცვლილებები და ა.შ ვერ ხორციელდება დეპარტამენტის უშუალო ჩართულობით. სწორედ ამიტომ გადაწყდა ისეთი cms-ის დამუშავება, რომელიც შეიძლება გამოყენებულ იქნას მრავალი დანიშნულებით (დეპარტამენტის, კონფერენციის და ა.შ) და იმართებოდეს ლოკალურად - დეპარტამენტის მიერ.

CMS წარმოადგენს უნივერსალურ პროგრამულ პროდუქტს, რომლის დახმარებითაც ხორციელდება ნებისმიერი სირთულის ვებ-გვერდის, დაწყებული მცირე საპრეზენტაციო ხასიათის ვებ-გვერდიდან – მრავალფუნქციური საინფორმაციო პორტალით დამთავრებული, ადმინისტრირება, განახლება და განვითარება სწრაფად და მარტივად. აუცილებელია, რომ მისი მოხერხებული და გასაგები ინტერფეისი საიტზე ინფორმაციის ეფექტური მართვის საშუალებას იძლეოდეს ყოველგვარი პროგრამისტებისა და html-ის სპეციალისტების დახმარების გარეშე.

დასმული ამოცანის გადასაწყვეტად ნაშრომში გამოყენებულია სხვადასხვა ვებ ტექნოლოგია: კერძოდ: HTML, CSS, Javascript ტექნოლოგიები ვებ-გვერდის ინტერფეისის შესაქმნელად, ბაზები - MYSQL, ხოლო ბიზნეს ლოგიკა (ვებ-გვერდის ძრავი) PHP framework - Laravel -ის გამოყენებით დაიწერა.

სამაგისტრო ნაშრომის ფარგლებში შეიქმნა ვებ-გვერდი, თავისი სამართავი სისტემით, საიდანაც იმართება ვებ-გვერდზე არსებული ნებისმიერი ფოტო-ვიდეო მასალა და ტექსტური ინფორმაცია რამდენიმე ენაზე. შესაძლებელია ვებ-გვერდიდან ელექტრონული წერილის გაგზავნა და ასევე მომავალი სტუდენტების ონლაინ რეგისტრაცია.

# CMS - Content Management System

## Abstract

The goal of the presented master work was to create a Content Management System to enable our department effectively represent their resources online, thus in frames of the work appropriate web page was created, that is devoted solely to Computer Science department and its programs. Tbilisi State University has an official web-site, but information displayed there is quite little and management of the page is done not by department itself. Thus, we decided to create a CMS, which can be used to provide information about our department's structure, professors, current teaching programs, trainings and conferences. The above mentioned cms should be convenient with user-friendly interface that enable department to modify and update its content effectively without the help of programmers.

In frames of the work concrete software – CMS was created based on various technologies: for front end and design some HTML, CSS, JavaScript was used, data storage is performed by MySQL database and back end is implemented using php framework - Laravel.

The web-site, with its content management system created, provides possibility to manage (insert/delete/update) any photo-video information and textual information on webpage on two different languages. It is also possible to send Email and handle registration process for future students online.

# შესავალი

## CMS - კონტენტის მართვის სისტემა

CMS-ის დანიშნულებაა ციფრული კონტენტის შექმნა და რედაქტირება. როგორც წესი, ის რამდენიმე მომხმარებლის ერთდროული მუშაობის საშუალებას იძლევა. CMS წარმოადგენს ფართოდ გავრცელებულ პროგრამულ პროდუქტს, რომელიც გამოიყენება საწარმოო კონტენტის მართვისა და ვებ კონტენტის მართვისთვის. CMS-ების უმეტესობა შეიცავს ვებ პუბლიკაციებს, ფორმატების კონტროლს, ისტორიის რედაქტირებას, ინდექსაციას, ძიებასა და ფილტრაციას. კონტენტის მართვის სისტემები კონტენტის დაყოფისა და პრეზენტაციის საშუალებას იძლევა.

ვებ კონტენტის მართვის სისტემა (WCM, WCMS) არის CMS, რომელიც შექმნილია ვებ-გვერდის მართვისათვის. ყველაზე ცნობილ CMS-ებს სწორედ WCMS წარმოადგენს. ვებ კონტენტი შეიცავს ტექსტებს, გრაფიკებს, ფოტოებს, ვიდეო-აუდიო მასალას, რუკებს, პროგრამულ კოდებს, რომლებიც გვიჩვენებს კონტენტს ან ამყარებს მომხმარებელთან კონტაქტს. ამ ყველაფრის მართვის საშუალებას ვებ-გვერდზე იძლევა WCMS.

როგორც წესი, CMS შედგება ორი ძირითადი ნაწილისგან. კონტენტის მართვის აპლიკაცია - CMA არის მომხმარებლის მხარის ინტერფეისი, რომელიც საშუალებას აძლევს თუნდაც გამოუცდელ მომხმარებელს დაამატოს, დაარედაქტიროს ან წაშალოს კონტენტი. კონტენტის მიწოდების აპლიკაცია (CDA) აკომპლირებს ამ ინფორმაციას და ანახლებს ვებ გვერდს. ციფრული მენეჯმენტის სისტემები არის კიდევ სხვა ტიპის CMS. ისინი მართავენ კონტენტს ზუსტად განსაზღვრული ავტორით ან მფლობელით, როგორებიცაა დოკუმენტები, ფილმები, სურათები, ტელეფონის ნომრები და მეცნიერული ინფორმაცია. კომპანიები ასევე იყენებენ CMS-ებს რომ შეინახონ, აკონტროლონ, გადახედონ და განახლონ დოკუმენტაცია.

## სხვადასხვა ტიპის ინფორმაციის მართვის სისტემები

შენი საქმისთვის მართვის სისტემის შერჩევისას (CMS) პოულობ მრავალ განსხვავებას სხვადასხვა ტიპის CMS-ებს შორის, მათი მახასიათებლების, ფუნქციების და ფასის მიხედვით.

არსებობს სამი ფართოდ გავრცელებული ტიპი: საჯარო, მესაკუთრეობრივი, პროგრამული უზრუნველყოფა, როგორც სერვისი. განვიხილოთ თითოეული მათგანი:

### საჯარო CMS:

საჯარო CMS-ები შეიგვიძლია გადმოვწეროთ საწყისი ფასის გარეშე. მათ არ აქვთ ლიცენზიის და განახლების გადასახადი ან კონტრაქტი. თუმცა შეიძლება მოგიწიოს გადახდა შემდეგი მომსახურებებისთვის:

- ტექნიკური დახმარება დაყენების დროს
- დაკონფიგურირება იმისთვის, რომ გააფართოვო ძირითადი შეთავაზება
- თავსებადი თარგები, მოდულები და plugin - ები (თუმცა ეს ყველაფერი უფასოც არსებობს)
- თანამშრომლების გადამზადება
- მხარდაჭერა, რომელიც მოიცავს პროგრამული უზრუნველყოფის მუდმივ განახლებას.

მსოფლიოში ყველაზე ფართოდ გავრცელებული საჯარო CMS-ებია:

- Wordpress
- Joomla
- Drupal
- Magento (e-commerce)
- Prestashop (e-commerce)

ეს არის მცირე ჩამონათვალი საჯარო CMS - ების, უფრო მეტი შეგიძლიათ ნახოთ [ბმულზე](#).

## მესაკუთრეობრივი

მესაკუთრეობრივი ან კომერციული CMS შექმნილია და იმართება კონკრეტული კომპანიის მიერ. მსგავსი CMS-ის გამოყენება ზოგადად მოიცავს:

- ლიცენზიის ყიდვას პროგრამული უზრუნველყოფის გამოსაყენებლად
- თვიურ ან წლიურ გადასახადს განახლებებისთვის და მხარდაჭერისთვის
- ასევე ასეთ შემთხვევაში ვიხდით დამატებით გადასახადს დაკონფიგურირებისთვის და სისტემური განახლებებისთვის, ისევე როგორც გადამზადებისთვის და მომავალი ტექნიკური, თუ სამომხმარებლო მხარდაჭერისთვის.

მსგავსი CMS-ებია:

- Kentico
- Microsoft sharepoint
- IBM enterprise content management
- Pulse cms
- Sitecore
- Shopify

სრული სია იხილეთ [ბმულზე](#).

მესაკუთრეობრივი CMS შეგიძლია გადააკეთო ჩაშენებული ფუნქციონალით, თუმცა ეს გამოიწვევს დამატებით ხარჯებს. თუ შესაძლებელია, უკეთესია რომ ვიპოვოთ ისეთი გადაწყვეტილება რომელიც ერგება ყველა ჩვენ მოთხოვნას.

თუ ვეძებთ CMS ახალი ბრენდის ვებ გვერდისთვის, საუკეთესო ვარიანტი იქნება ისეთი cms ის არჩევა რომელსაც აქვს ყველა მახასიათებელი და თავსებადია არა მხოლოდ მიმდინარე გეგმებისთვის არამედ სამომავლო ბიზნეს საჭიროებებისთვის.



## პროგრამული უზრუნველყოფა, როგორც სერვისი (SAAS) CMS

SAAS CMS ზოგადად მოიცავს ვებ კონტენტის მართვის სისტემას, ვებ ჰოსტინგს და ტექნიკურ მხარდაჭერას ერთი მომწოდებლისგან. ეს არის ვირტუალური სერვერები რომლებიც არის ქლაუდზე და დაფუძნებულია სააბონენტო მოდელის მიხედვით, როგორც წესი თითო მომხმარებელი ან თითო ვებ გვერდი. ფასი როგორც წესი მოიცავს:

- ინფორმაციის ტრანსფერების რაოდენობას (bandwidth ვებ გვერდიდან და ვებ გვერდის მიმართულებით)
- კონტენტის და ინფორმაციის საცავს
- მომავალ მხარდაჭერას

არსებობს ორი ტიპის Cloud კონტენტის მართვის სისტემა:

- 'Fully cloud' cms ხშირად მოდის პაკეტის ან სერვისის ნაწილი. ტიპიურად, არსებობს მესაკუთრეობრივი სისტემები მიმწოდებლის კონტროლის ქვეშ, ამიტომ ყოველთვის არ არის შესაძლებელი შეიცვალოს ფუნქციონალი ჩვენი საჭიროებების მიხედვით.
- 'Partial cloud' cms განთავსებულია საკუთარ cloud ვებ სერვერზე. ის ფუნქციონალის შეცვლის საშუალებას იძლევა დამატებითი მოდულებითა და კოდის ცვლილებით.

Cloud CMS გვთავაზობს რამდენიმე მნიშვნელოვან უპირატესობას პატარა და საშუალო ზომის ბიზნესებისთვის. მაგალითად:

- ფასები, როგორც წესი, არის დაბალი - დაბალი ინსტალაციის საფასური, როგორც წესი, ფარავს საწყის იმპლემენტაციას
- SAAS მომწოდებელი გთავაზობს განახლებებს, მხარდაჭერასა და ტექნიკური საკითხების მოგვარებას.
- პროგრამული უზრუნველყოფა ხელმისაწვდომია ნებისმიერი კომპიუტერით, ლეპტოპით ან ტელეფონით ინტერნეტით

- განახლებები ხდება რეალურ დროში
- პაკეტები ადვილად სკალირდება - შესაძლებელია მეტი ვებ-გვერდის ან მომხმარებლის დამატება საჭიროებების შესაბამისად.

## MYSQL

MySQL არის ღია რელაციური მონაცემთა ბაზის მართვის სისტემა. მისი სახელი შედგენილია “My” (თანადამფუძნებლის Michael Widenius-ის ქალიშვილის სახელი) და “SQL” Structured Query Language - ის აბრევიატურა.

MySQL არის უფასო და ღია პროგრამული უზრუნველყოფა ზოგადი საჯარო გამოყენების ლიცენზიის ქვეშ და ასევე ხელმისაწვდომია სხვადასხვა საკუთრების ლიცენზიები. MySQL ეკუთვნოდა და სპონსორდებოდა შვედური კომპანიის მიერ MySQL AB, რომელიც იყიდა Sun Microsystems-მა 2010 წელს, როდესაც Oracle შეიძინა Sun-მა, Widenius-მა გამოიყენა MySQL პროექტი MariaDB-ს შესაქმნელად.

MySQL არის LAMP ვებ აპლიკაციის პროგრამული უზრუნველყოფის სტეკის კომპონენტი, რომელიც არის აკრონიმი (ხმოვანი ბგერითი შემცირება) ლინუქსისთვის, აპაჩისთვის, MySQL-თვის, PHP/Perl/Python-თვის. MySQL-ს იყენებს ბევრი ბაზაზე ორიენტირებული ვებ აპლიკაცია, მაგალითად: Droopal, Joomla, phpBB და WordPress. MySQL-ს იყენებს ასევე ბევრი პოპულარული ვებ გვერდი, როგორებიცაა: Facebook, Flickr, MediaWiki, Twitter და Youtube.

MySQL დაწერილია C და C++ ზე და მისი SQL პარსერი დაწერილია yacc-ში. MySQL მუშაობს ბევრ პლატფორმაზე, როგორებიცაა: AIX, BSDi, FreeBSD, HP-UX, eComStation, Linux, macOS, Windows და სხვა. MySQL ის პორტი OpenVMS ში ასევე არსებობს.

## ვებ ტექნოლოგიები

განვიხილოთ პროგრამული ენები და ტექნოლოგიები, რომლებიც გვეხმარება შევქმნათ ვებ გვერდი. ვებ ტექნოლოგიები იყოფა ორ ნაწილად: საიტის ინტერფეისის ასაწყობად საჭირო ენები და ძრავის შესაქმნელი ე.წ სერვერული ენები. საიტის ინტერფეისის ასაწყობად ნაშრომისთვის გამოვიყენებთ : HTML - ვებ გვერდის მარკირების ენა , CSS - მარკირებული ვებ-გვერდის სტილებით გაფორმების ენა, JavaScript - საიტის ინტერაქტივისა და ეფექტების

შემუშავების ენა. განვიხილოთ ენები, რომლების გამოყენებაც შეიძლება ვებ-გვერდის ძრავის შესაქმნელად:

## Ruby

Ruby არის ინტერპრეტირებული, მაღალი დონის, ზოგადი გამოყენების პროგრამული ენა. იგი შეიქმნა 1990 წლის შუა ხანებში. მისი შემქმნელია იაპონელი Yukihiro "Matz" Matsumoto.

Ruby არის დინამიური ტიპის (მაღალი დონის კლასის პროგრამული ენა, რომელიც გაშვების დროს ასრულებს ძირითად პროგრამულ დავალებებს, რომლებსაც სტატიკური ენები კომპილაციის დროს ასრულებენ) და garbage-collected (GB - ფორმაა რომელიც მუხსიერებას ავტომატურად მართავს და ამენეჯმენტებს). მას აქვს სხვადასხვა პროგრამული პარადიგმის მხარდაჭერა, რომელიც მოიცავს პროცედურულ, ობიექტზე ორიენტირებულს და ფუნქციურ პროგრამირებას. როგორც შემქმნელები აღნიშნავენ, Ruby-ში შეიმჩნევა ისეთი ენების გავლენა როგორებიცაა : Perl, Smalltalk, Eiffel, Ada, Basic, and Lisp.

## Java

Java არის კლასებზე დაფუძნებული, ობიექტზე ორიენტირებული, ზოგადი გამოყენების პროგრამული ენა და შექმნილია იმისთვის, რომ ჰქონდეს რაც შეიძლება ცოტა იმპლემენტაცია. იგი განკუთვნილია იმისთვის, რომ აპლიკაციის დეველოპერებს შეეძლოთ დაწერონ ერთხელ და გაუშვან ყველგან (write once, run anywhere (WORA)), რაც ნიშნავს იმას, რომ Java-ს კოდი ერთხელ რომ დაკომპილირდება შეიძლება გაეშვას ყველა პლატფორმაზე, რომელსაც აქვს Java-ს მხარდაჭერა ახლიდან კომპილაციის გარეშე. ჯავას აპლიკაციები არის ტიპიურად დაკომპილირებული ბაიტ კოდში, რომელიც შეიძლება გაეშვას ნებისმიერ Java-ს ვირტუალურ ძრავზე კომპიუტერის არქიტექტურის მიუხედავად. მისი სინტაქსი C და C++ ენების მსგავსია, მაგრამ ამას აქვს უფრო დაბალი დონის ობიექტები ვიდრე ზემოთ ხსენებულ ენებს. 2018-ში Java იყო ერთ-ერთი ყველაზე პოპულარული პროგრამული ენა Github-ის მონაცემებით, განსაკუთრებით კლიენტ-სერვერის ვებ აპლიკაციებში.

## **.NET Framework**

.NET framework ("dot net") არის framework, შექმნილი Microsoft-ის მიერ, რომელიც ეშვება პირდაპირ Microsoft Windows - ზე. ის მოიცავს დიდი ზომის კლასების ბიბლიოთეკას, რომელსაც ჰქვია Framework Class Library (FCL) და უზრუნველყოფს ენების თავსებადობას სხვადასხვა ენებთან. პროგრამები, რომლებიც არის .NET-თვის ეშვება პროგრამულ გარემოში, რომელსაც ჰქვია Common Language Runtime (CLR). CLR არის აპლიკაციის ვირტუალური ძრავი, რომელიც უზრუნველყოფს სერვისებს, როგორებიცაა უსაფრთხოება, მეხსიერების მენეჯმენტი და exception-ების გატარება. FCL და CLR ერთდროულად წარმოადგენს .NET Framework-ს. FCL უზრუნველყოფს მომხმარებლის ინტერფეისს, ინფორმაციაზე წვდომას, ბაზასთან კავშირს, კრიპტოგრაფიას, ვებ აპლიკაციის დეველოპმენტს, რიცხვების ალგორითმებს და ქსელთან კავშირს. პროგრამისტები პროგრამული უზრუნველყოფას ქმნიან მათი კოდების კომბინაციით .NET-თან და სხვა ბიბლიოთეკებთან. Framework განკუთვნილია ახალი აპლიკაციების გამოყენებისთვის რომელიც შექმნილია ვინდოუსისთვის. Microsoft-მა ასევე შექმნა ინტეგრირებული დეველოპმენტის გარემო .NET-თვის, რომელსაც ჰქვია Visual Studio.

## **Python**

Python არის ინტერპრეტირებული, მაღალი დონის, ზოგადი გამოყენების პროგრამული ენა, რომელიც შექმნა Guido van Rossum-მ 1991 წელს. მისი ენის კონსტრუქცია და ობიექტზე ორიენტირებული მიდგომა მიზანმიმართულია იმისთვის, რომ პროგრამისტებმა წერონ სუფთა და ლოგიკური კოდი პატარა თუ დიდი პროექტებისთვის.

Python არის დინამიური ტიპის (მაღალი დონის კლასის პროგრამული ენა, რომელიც გაშვების დროს ასრულებს ძირითად პროგრამულ დავალებებს, რომლებსაც სტატიკური ენები კომპილაციის დროს ასრულებენ) და garbage-collected (GB - ფორმა რომელიც მეხსიერებას ავტომატურად მართავს და ამენეჯმენტებს). მას აქვს სხვადასხვა პროგრამული პარადიგმის მხარდაჭერა, რომელიც მოიცავს პროცედურულ, ობიექტზე ორიენტირებულს და ფუნქციურ პროგრამირებას. Python ხშირად აღიწერება როგორც "batteries included" ენა მისი სრულყოფილი სტანდარტული ბიბლიოთეკის გამო.

## PHP

**PHP** (PHP: Hypertext Preprocessor) — პროგრამირების ენა, რომელიც თავისი ოპერატორებით PHP კოდს, ბრაუზერში გარდაქმნის HTML და XHTML კოდად. ბრაუზერის მისამართების პანელში საიტის გამოძახებისას სახელწოდების ბოლოს php დაბოლოება ნიშნავს იმას, რომ მოცემული საიტი შექმნილია PHP კოდით. ბრაუზერში გვერდის ვებ-კოდების მოთხოვნისას, არ გამოჩნდება PHP კოდი, გამოჩნდება მხოლოდ HTML კოდი. ეს ტექნოლოგია პროგრამირებაში ახალი არაა და იგი ჯერ კიდევ PERL-ში გამოიყენებოდა, რომლისაგანაც წარმოიშვა PHP. 1994 წელს რასმუს ლერდორფმა, რომელსაც სურდა ხსენებული ხერხის განხორციელება ბრაუზერში, შექმნა პირველი ვერსია, რომელიც ამავე წელს გამოაქვეყნა, სახელწოდებით Personal Homepage Tools. პირველი ვერსია მალე გადამუშავდა და მას ოფიციალურად დაერქვა PHP.

არცთუ დიდი ხნის წინ Web-კვანძებიდან მომხმარებელს ადგილზე ძირითადად სტატიკური HTML-ფაილები გადმოეცემოდა, რომელთაც იგი ბრაუზერის მეშვეობით მხოლოდ კითხულობდა. დღეს კი WEB-სივრცეში უმეტესწილად დინამიკური Web-ფურცლები ანუ DHTML-ფაილები გადაიგზავნება. მათი აწყობა ხდება Web-კვანძების სერვერზე განთავსებული WEB-პროგრამების გაშვებით, რასაც მომხმარებელი ახორციელებს საკუთარი კომპიუტერიდან სისტემასთან დიალოგში. ინტერნეტში პროგრამების შექმნისათვის დღეს ყველაზე პოპულარული WEB-დაპროგრამების ენა არის PHP4. იგი განსაკუთრებით აადვილებს სერვერებზე განლაგებულ მონაცემთა ბაზებთან ურთიერთობას. დასაწყისში PHP შეიქმნა, როგორც მაკროსების კრებული, მარტივი სახის პირადი WEB-ფურცლების ფორმირებისთვის. შემდეგ კი იგი იქცა სრულფასოვან დაპროგრამების ენად, რომელზეც დაწერილი კონსტრუქციები ჩაიდგმება სერვერზე მოთავსებულ HTML- ტექსტებში. WEB-ფურცლის გამოძახებისას PHP-ის პრეპროცესორის მიერ ადგილზე ხდება ამ კონსტრუქციების (პროგრამული ფრაგმენტების) დამუშავება და HTML-ის არსებულ ფრაგმენტებს შორის ჩაერთება შესაბამისი ინფორმაცია, მაგალითად, ბაზიდან ამოღებული მონაცემები, სერვერზე კონსტრუირებული გრაფიკული გამოსახულებები და სხვ. PHP-ის ვერსია - PHP4(5) - მისაერთებელი მოდულის სახით კომპილირდება დღეისათვის ყველაზე პოპულარულ WEB-

სერვერ-პროგრამა Apache-სთვის. სერვერზე PHP-პროგრამები უმეტესწილად ურთიერთობენ MySQL მონაცემთა ბაზასთან (რომელიც უფასოდ ვრცელდება). ასე რომ, APACHE, MySQL და PHP ტრიადას დღევანდელ დღეს კონკურენტი არ ჰყავს. თუმცა, ცხადია, PHP-ს მუშაობა შეუძლია სხვა WEB-გარემოში და სხვა ბაზებთანაც. საერთოდ, PHP-პროგრამებს ინტერპრეტატორი ამუშავებს, მაგრამ შესაძლებელია მათი კომპილირებაც (თუმცა ეს შესაძლებლობა მხოლოდ კომერციული დამატების სახით არსებობს), რაც კიდევ უფრო ამაღლებს სისტემის მწარმოებლურობას. CGI-სთან შედარებით PHP-ს ის მნიშვნელოვანი უპირატესობა გააჩნია, რომ მისი ინტერპრეტატორი ფუნქციონირებს სერვერთან მჭიდრო ინტეგრაციის პირობებში საჭირო არ არის თითოეული პროგრამის შესრულებისას სისტემის ხელახლა გაშვება, რაც მნიშვნელოვნად ამაღლებს მუშაობის სისწრაფეს. PHP-ს რიგი სხვა ღირსებებიც გააჩნია:

- პროდუქტის ღიაობა;
- პროგრამების დაწერის გაადვილება HTML-ტექსტისა და თვით პროგრამული ნაწილის განცალკევების გამო;
- პროგრამების ერთი ოპერაციული სისტემიდან სხვაში გადატანის სიმარტივე;
- პროგრამების შესრულების საკმარისი სისწრაფე.

## პოპულარული PHP ფრეიმვორქები

საბაზო php ენაზე წერა დაკავშირებულია ისეთ სირთულეებთან, როგორცაა დაცულობა და პროექტზე მუშაობის სისწრაფე. მსგავსი პრობლემების გადასაჭრელად შექმნილია სხვასხვა php framework. PHP framework უზრუნველყოფს აპლიკაციის ძირითადი სტრუქტურის შექმნას და განაპირობებს პროექტის სწრაფ განვითარებას. ის უზრუნველყოფს საბაზისო სტრუქტურას, რომელიც ამარტივებს ვებ აპლიკაციის შექმნის პროცესს.

### Laravel

Laravel შეიქმნა 2011 წელს და გახდა ყველაზე პოპულარული უფასო ღია კოდის framework მსოფლიოში. იმიტომ, რომ მისი საშუალებით შესაძლებელია კომპლექსური ვებ აპლიკაციების უფრო მარტივად და სწრაფად შექმნა სხვა framework -ებთან შედარებით. Laravel ამარტივებს

პროგრამირების პროცესს ისეთი ზოგადი ამოცანების გამარტივებით როგორცაა როუტინგი, სესიები, ქეშირება და აუდენტიფიკაცია.

### **რატომ ვიყენებთ ლარაველს**

- Laravel არის შესაფერისი, როდესაც ხდება კომპლექსური აპლიკაციების ბექენდის დეველოპმენტი, დიდი იქნება თუ პატარა. დასაყენებლად იგი ძალიან მარტივია ოფიციალური დოკუმენტაციის გამოყენებით.
- იგი არის PHP-ის Framework ყველა შესაძლებლობით, რომელიც გეხმარება კომპლექსური აპების შექმნაში. მათ შორის: ინფორმაციის მიგრაცია, MVC არქიტექტურის მხარდაჭერა, უსაფრთხოება, როუტინგი, ვიუები და ავტორიზაცია და სხვა.
- Laravel არის ძალიან ექსპრესიული და მისი სისწრაფე და დაცულობა ამართლებს თანამედროვე ვებ აპლიკაციების მოლოდინებს. დეველოპერებისთვის ვისაც უნდათ ააწყონ B2B და საწარმო ვებ გვერდები, რომლებიც იცვლება ვებ ტრენდების ცვლილებასთან ერთად, ლარაველი არის სწორი არჩევანი.

## **CodeIgniter**

CodeIgniter ცნობილია მისი დაბალი მოცულობით. იგი არის სულ რაღაც 2MB დოკუმენტაციით. CodeIgniter არის PHP framework დინამიური ვებ გვერდების შესაქმნელად. იგი გვთავაზობს სხვადასხვა ჩაშენებულ მოდულებს, რომლებიც გეხმარება ძლიერი და ხელმეორედ გამოყენებისთვის ვარგისი კომპონენტების შექმნაში.

### **რატომ ვიყენებთ CodeIgniter - ს**

- CodeIgniter არის მსუბუქი და ძლიერი PHP framework, რომელიც შეუფერხებლად ყენდება სხვა framework-ებისგან განსხვავებით. მარტივი ინსტალაციის პროცესისა და კარგად წარმოდგენილი დოკუმენტაციის გამო იგი არის იდეალური დამწყებებისთვის.
- საკვანძო მახასიათებლები მოიცავს MVC არქიტექტურას, შეცდომების დაჭერას, ჩაშენებულ უსაფრთხოებას და მარტივ და შესანიშნავ დოკუმენტაციას.

- სხვა framework-ებთან შედარებით, CodeIgniter არის შესამჩნევად სწრაფი. იგი ასევე გვთავაზობს მყარ პერფორმანსს, ის არის კარგი არჩევანი, როდესაც გჭირდება მსუბუქი აპლიკაცია. **ერთი გაფრთხილება:** CodeIgniter არ არის კარგი არჩევანი, როდესაც აპლიკაციას სჭირდება მაღალი დონის დაცვა.

## Symfony

Symfony framework გაეშვა 2005 წელს, მაგრამ ის უფრო ადრეც არსებობდა სხვა framework-ებისგან განსხვავებით, იგი არის საიმედო და სანდო პლატფორმა. Symfony არის ფართო გამოყენების PHP MVC Framework და ერთადერთი ცნობილი framework, რომელიც მიყვება PHP და ვებ სტანდარტებს.

### რატომ ვიყენებთ Symfony - ს

- Symfony არის საუკეთესო არჩევანი დიდი ზომის პროექტების დეველოპმენტისთვის. მარტივია მისი დაყენება და კონფიგურირება სხვადასხვა პლატფორმაზე.
- Symfony არის ასევე მორგებადი და შეიძლება უფრო დიდ პროექტებში ინტეგრაცია, როგორცაა მაგალითად Drupal, Symfony-ს და Laravel-ს აქვს ბევრი მსგავსი და უნიკალური მახასიათებელი, რომელიც ართულებს იმის გარკვევას თუ, რომელი უფრო კარგი framework-ია.

## MVC (model-view-controller)

**Model-view-controller** (ცნობილი როგორც MVC), არის მომხმარებლის ინტერფეისის დეველოპმენტისთვის ფართოდ გამოყენებადი არქიტექტურული ნიმუში, რომელიც აპლიკაციას ყოფს სამ დაკავშირებულ ნაწილად. იგი შეიქმნა იმისთვის, რომ გაყოფილიყო გზები ინფორმაციის მიღების და წარმოდგენის მომხმარებლისთვის.

ტრადიციულად გამოიყენება მომხმარებლის გრაფიკული ინტერფეისისთვის (guis), ეს არქიტექტურა პოპულარულია ვებ აპლიკაციების შესაქმნელად. პოპულარული პროგრამული ენები როგორცაა JavaScript, Python, Ruby, PHP, Java, და C# აქვთ mvc framework-ები, რომლებიც გამოიყენება ვებ აპლიკაციების დეველოპმენტში.



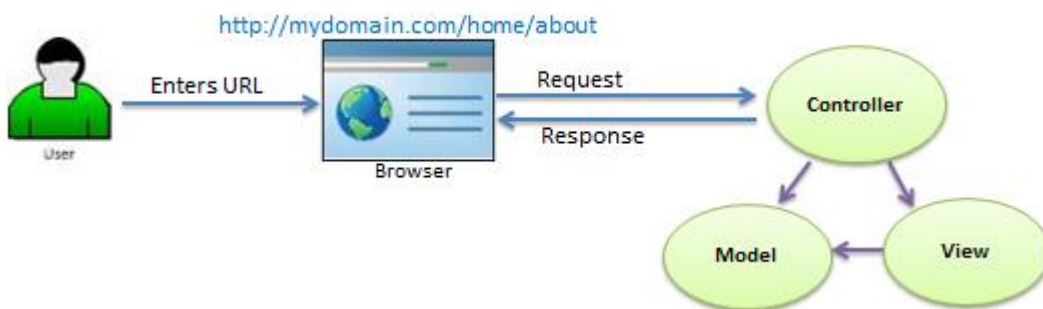
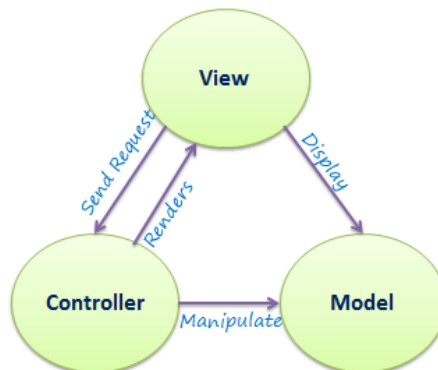
## კომპონენტები

**Model** - ცენტრალური კომპონენტი. ის არის აპლიკაციის დინამიური ინფორმაციული სტრუქტურა, მომხმარებლის ინტერფეისიდან დამოუკიდებლად. ის პირდაპირ მართავს აპლიკაციის ინფორმაციას, ლოგიკასა და წესებს.

**View** - ინფორმაციის ნებისმიერი წარმოდგენა, როგორცაა ჩარტი, დიაგრამა ან ცხრილი. სხვადასხვა view ერთი და იგივე ინფორმაციითაც არის შესაძლებელი.

**Controller** - იღებს შემავალ ინფორმაციას და გადაჰყავს ბრძანებებში მოდელისთვის და view - სთვის.

სურათზე წარმოდგენილია ინტერაქცია model-ს, view-სა და controller-ს შორის:



## დასმული ამოცანის გადასაჭრელად დაგეგმილი ეტაპები:

სამაგისტრო ნაშრომის მიზანია კომპიუტერული მეცნიერებების დეპარტამენტისთვის ვებ-გვერდისა და მისი კონტენტის მართვის სისტემის შექმნა. ვებ-გვერდის შესაქმნელად შეირჩა პროგრამული ენები: ინტერფეისისთვის - HTML, CSS, JavaScript და ძრავისთვის - PHP framework - Laravel. დასახული მიზნის სწორად შესრულებისთვის გაიწერა სამუშაო გეგმა:

**საიტის ტექნიკური დავალების დაწერა** - დაწვრილებით ყველა მოდულის აღწერა, რომელიც საჭიროა გამოჩნდეს საიტზე, თითოეული მოდულისთვის რა ველების მართვაა საჭირო ადმინისტრატორის მხრიდან.

**საიტის ინტერფეისის აწყობა** - ერთმანეთისგან განსხვავებული გვერდების აწყობა HTML, CSS, JavaScript ტექნოლოგიების გამოყენებით. კვლევის შედეგად დადგინდა, რომ საჭიროა შემდეგი გვერდების აწყობა:

- მთავარი გვერდი
- სიახლეების ჩამონათვალი
- აკადემიური პერსონალის გვერდი
- ვაკანსიების გვერდი
- გაცვლითი პროგრამების გვერდი
- ტექსტური გვერდები
- კონტაქტის გვერდი
- რეგისტრაციის ფორმის გვერდი

**საიტის სამართავი პანელის ინტერფეისის აწყობა** - მარტივად გასაგები, სუფთა დიზაინის აწყობა, იმისათვის რომ ვებ ტექნოლოგიაში გაურკვეველი ადმინისტრატორისთვისაც ძალიან მარტივი იყოს საიტის მართვა.

- მთავარი გვერდი
- ჩამონათვალის ინტერფეისი
- რედაქტირებისა და დამატების ფორმების ინტერფეისი

**ძრავის აწყობა**

- საიტის სტრუქტურის შესაბამისი ცხრილების შექმნა
- მართვის პანელში ყველანაირი ინფორმაციის მარტივი ინტერფეისით გამოტანა, სამართავი პანელიდან ბაზაში ახალი ინფორმაციის დამატება, არსებული ინფორმაციის რედაქტირების და წაშლის ფუნქციონალი, სორტირების ფუნქციონალის დაწერა, ველების თარგმნის ფუნქციონალი
- ბაზაში არსებული ინფორმაციის გამოტანა საიტის ინტერფეისში
- საიტის ინტერფეისიდან მეილის ფორმის ფუნქციონალის გაკეთება
- რეგისტრაციის ფორმის გაგზავნა და გაგზავნილი ფორმების მართვის პანელში გამოჩენა

**სერვერზე გამართვა** - გამზადებული ვებ-გვერდისა და მისი სამართავი პანელის სერვერზე ატვირთვა და გამართვა

# ვებ გვერდის ტექნიკური დავალება:

საიტი უნდა იყოს ორენოვანი: ქართული, ინგლისური

## 1. ტექსტური გვერდები:

- ჩვენ შესახებ
  - სათაური
  - აღწერა (ტექსტური რედაქტორი)
  - ორი მთავარი სურათი
- სასწავლო პროცესი თსუ-ში
  - აღწერა (ტექსტური რედაქტორი)
- სასწავლო ცხრილები
  - მეცადინეობის განრიგის ასატვირთი ველი
  - კონსტულტაციების განრიგი ასატვირთი ველი
  - გამოცდების განრიგი
- სამეცნიერო წრე
  - აღწერა
- აბიტურიენტებისთვის
  - აღწერა

## 2. სიახლეები

- სათაური
- აღწერა

- მთავარი სურათი
- დამატების თარიღი
- გამოქვეყნებული იყოს თუ არა

### 3. სლაიდერი

- სათაური
- აღწერა
- ბმული
- მთავარი სურათი
- გამოქვეყნებული იყოს თუ არა
- რიგითობის შეცვლა

### 4. პროგრამები

- სათაური
- ხელმძღვანელი
- აღწერა
- ტიპი
  - საბაკალავრო
  - სამაგისტრო
  - სადოქტორო
- სურათი
- გამოქვეყნებული იყოს თუ არა

5. სტრუქტურა

- სათაური
- პოზიცია
- აღწერა
- სურათი
- გამოქვეყნებული იყოს თუ არა

6. თანამშრომლები

- სათაური
- აღწერა
- ტიპი
  - აკადემიური პერსონალი
  - დამხმარე პერსონალი
- სურათი
- გამოქვეყნებული იყოს თუ არა

7. მრჩეველთა საბჭო

- სათაური
- აღწერა
- სურათი
- გამოქვეყნებული იყოს თუ არა

8. კვლევები

- სათაური
- აღწერა
- სურათი
- გამოქვეყნებული იყოს თუ არა

9. პროექტები

- სათაური
- აღწერა
- დამატების თარიღი
- სურათი
- გამოქვეყნებული იყოს თუ არა
- გამოქვეყნდეს თუ არა სტუდენტურ პროექტებში

10. ღონისძიებები

- სათაური
- აღწერა
- დამატების თარიღი
- სურათი
- გამოქვეყნებული იყოს თუ არა

11. კარიერა

- ვაკანსიები
  - სათაური

- მდებარეობა
- აღწერა
- ტიპი
  1. სრული
  2. ნახევარი
  3. რემოუთ
- გამოქვეყნებული იყოს თუ არა

○ სტაჟირება

- სათაური
- აღწერა
- ტიპი
  1. სტაჟირება
  2. კვალიფიკაციის ამაღლება

12. საკონტაქტო ინფორმაცია

- მისამართი
- ტელეფონი
- ელ. ფოსტა
- რუკა
  - გრძედი
  - განედი



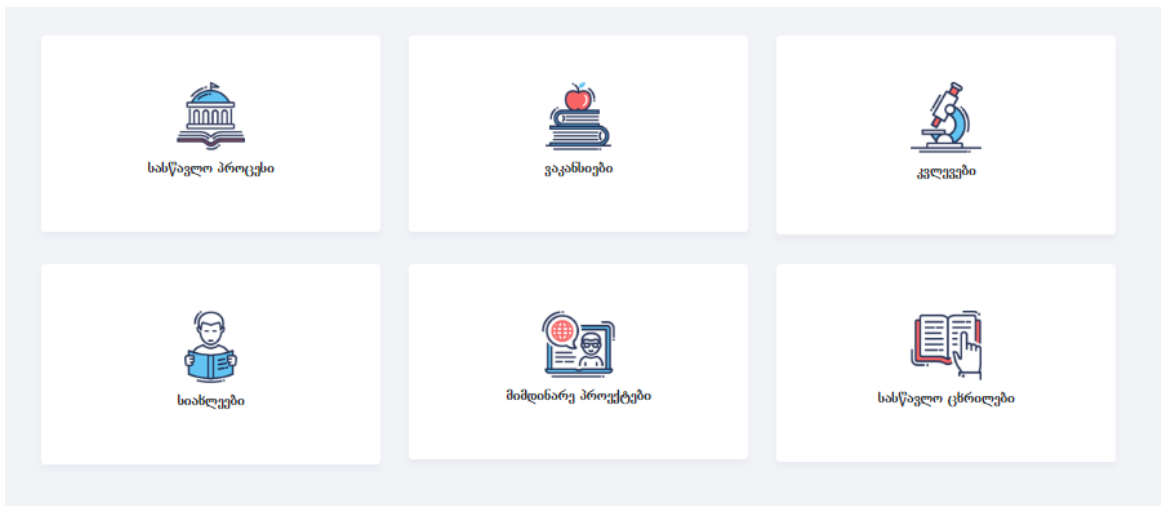
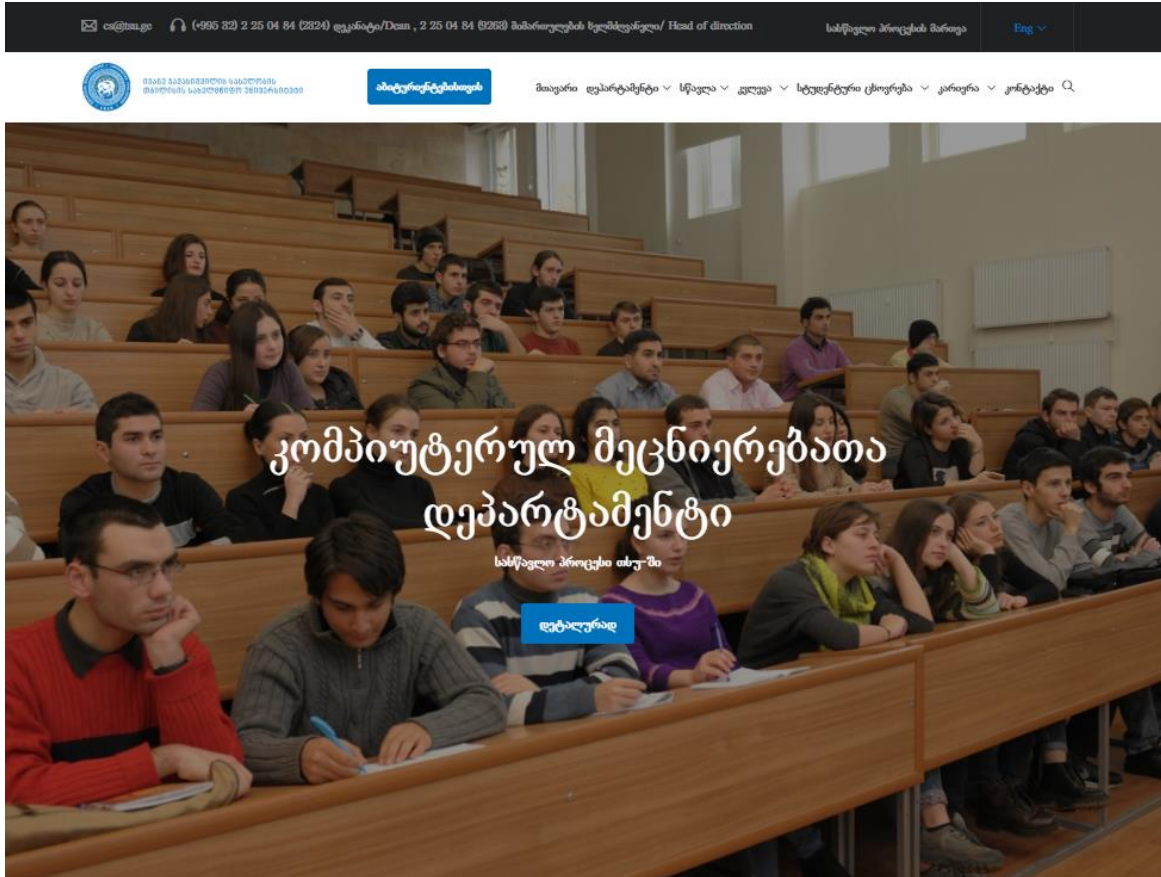
- Facebook ბმული
  - Instagram ბმული
13. ადმინისტრატორები
- ნიკი
  - ელ. ფოსტა
  - პაროლი
14. საკონტაქტო ფორმა (კონტაქტის გვერდზე გამოჩნდეს ფორმა, რომელიც გაიგზავნება ადმინისტრატორის ელექტრონულ ფოსტაზე)
- სახელი
  - ელ. ფოსტა
  - ტელეფონის ნომერი
  - შეტყობინება
15. უცხო ქვეყნის მოქალაქეებისთვის თსუ-ში ჩარიცხვისათვის შესავსები ონლაინ აპლიკაციის ფორმა, გაგზავნილი ფორმა უნდა გამოჩნდეს ადმინ პანელში ჩამონათვალის სახით.
- სახელი
  - გვარი
  - სქესი
  - პირადი ნომერი
  - დაბადების თარიღი
  - საცხოვრებელი ქვეყანა, ქალაქი

- მისამართი
- ელ. ფოსტა
- ტელეფონი
- ფაილების ასატვირთი ველები
  - სკოლის დამამთავრებელი დიპლომის ორიგინალი
  - transcript of academic record
  - პასპორტის ასლი
  - ინგლისურის ცოდნის საერთაშორისოდ დამოწმებული სერტიფიკატი

# ამოცანის პრაქტიკული რეალიზაცია

## საიტის ინტერფეისის აწყობა

### მთავარი გვერდი



## სიანსლები



28 ივნისი 2019

### “შუვანე ყუთი” უნივერსიტეტში

სტუდენტური ინიციატივით სახელმწიფო უნივერსიტეტის კომპიუტერული მიმართულების დეპარტამენტი ჩაერთო პროექტ “შუვანე ყუთში” პროექტის ფარგლებში უნივერსიტეტის XI კორპუსის ტერიტორიაზე განტავსდ...

[დეტალურად](#)



28 ივნისი 2019

კომპიუტერული მეცნიერებების მიმართულების სტუდენტთა სამეცნიერო წრის სსდომა



19 ივნისი 2019

კომპიუტერული მეცნიერებების მიმართულების სტუდენტთა სამეცნიერო წრის სსდომა



19 ივნისი 2019

კომპიუტერული მეცნიერებების მიმართულების სტუდენტთა სამეცნიერო წრის სსდომა



ივანე ჯავახიშვილის სახელობის  
თბილისის სახელმწიფო უნივერსიტეტი

### საკონტაქტო ინფორმაცია

☎ (+995 32) 2 25 04 84 (2824) დეკანი/Dean , 2 25 04 84 (8263) მიმართულების ხელმძღვანელი/ Head of direction  
 ✉ [ca@tsu.edu.ge](mailto:ca@tsu.edu.ge)  
 📍 თბილისი, ი.ჭავჭავაძის გამზ. 5, 0179, იმე, II კორპუსი.

### საჭირო ბმულები

დეპარტამენტის ისტორია  
 კონტაქტი  
 სიანსლები  
 სტუდენტური ღონისძიებები  
 პროექტები



.unic\*de

# სიახლეების ჩამონათვალი

თქვენი უცხოელი (+995 82) 2 25 04 84 (2824) დეკანი/Dean, 2 25 04 84 (2823) მმართველის ხელმძღვანელი/ Head of direction

საქართველოს პროექტის მართვა Eng


თბილისის სახელმწიფო უნივერსიტეტი

ამატორებისთვის

მთავარი დეპარტამენტი სწავლა კულტურა სტუდენტური ცხოვრება კარიერა კონტაქტი


მთავარი >> სიახლეები

## სიახლეები



"შვებავ ყუთი" უნივერსიტეტში


28 ივნისი 2019



Cloud Computing

კომპიუტერული მეცნიერებების მმართველის სტუდენტთა სამეცნიერო წრის სხდომა

28 ივნისი 2019




თბილისის სახელმწიფო უნივერსიტეტი

IVANE JAVAKHISHVILI  
TBILISI STATE UNIVERSITY


კომპიუტერული მეცნიერებების მმართველის სტუდენტთა სამეცნიერო წრის სხდომა

19 ივნისი 2019




თბილისის სახელმწიფო უნივერსიტეტი

IVANE JAVAKHISHVILI  
TBILISI STATE UNIVERSITY



www.bgu.ge



# აკადემიური პერსონალის გვერდი



მეცნიერებათა საბუნების  
მეცნიერებათა საბუნების 360326180308

ამიტორიტეტისთვის

მთავარი [დეპარტამენტი](#) ▾ [სწავლა](#) ▾ [კვლევა](#) ▾ [სტუდენტური ცხოვრება](#) ▾ [კარიერა](#) ▾ [კონტაქტი](#) 🔍

მთავარი » [დეპარტამენტის სტრუქტურა](#)

## დეპარტამენტის სტრუქტურა

### დეპარტამენტის სელმძღვანელი

პროფ. მანანა ხაჩიძე

### განათლების ხარისხის სამსახურის წარმომადგენელი

ასოც. პროფ. ნათელა ბრწყინიძე

### თეორიული ინფორმატიკის კათედრა

კათედრის გამგე – პროფ. ალექსანდრე  
გამყრულაძე

### ტექნიკური ინფორმატიკის კათედრა

მანანა ხაჩიძე

### პრაქტიკული ინფორმატიკის კათედრა

ასოც. პროფ. ირინა სუციშვილი

### გამოყენებითი ინფორმატიკის კათედრა

კათედრის გამგე – პროფ. ვია სარხილაძე

### სასწავლო კვლევითი ლაბორატორია

ლაბორატორიის სელმძღვანელი –  
ჯულიეტა გაბლოშვილი



მეცნიერებათა საბუნების  
მეცნიერებათა საბუნების 360326180308

ამიტორიტეტისთვის

მთავარი [დეპარტამენტი](#) ▾ [სწავლა](#) ▾ [კვლევა](#) ▾ [სტუდენტური ცხოვრება](#) ▾ [კარიერა](#) ▾ [კონტაქტი](#) 🔍

მთავარი » [დეპარტამენტის სტრუქტურა](#) » [დეპარტამენტის სელმძღვანელი](#)

## დეპარტამენტის სტრუქტურა



### დეპარტამენტის სელმძღვანელი

პროფ. მანანა ხაჩიძე

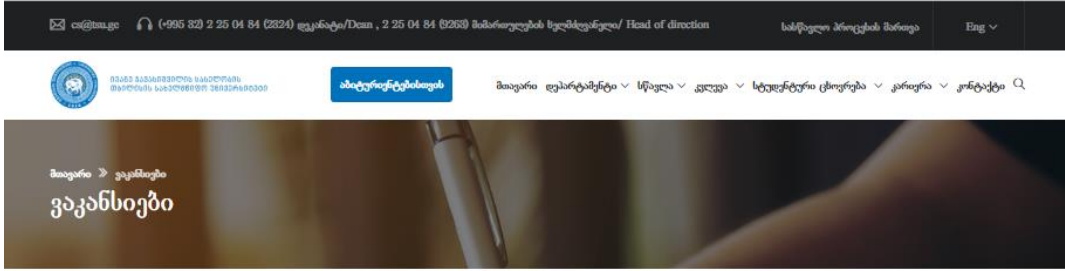
საკონტაქტო ინფორმაცია:

ოფისი: თსუ შე-11 კორპ. ოთახი 324

ტელეფონი: +995 577 295073

ელ-ფოსტა: [manana.khachidze@tsu.ge](mailto:manana.khachidze@tsu.ge)

# ვაკანსიების გვერდი



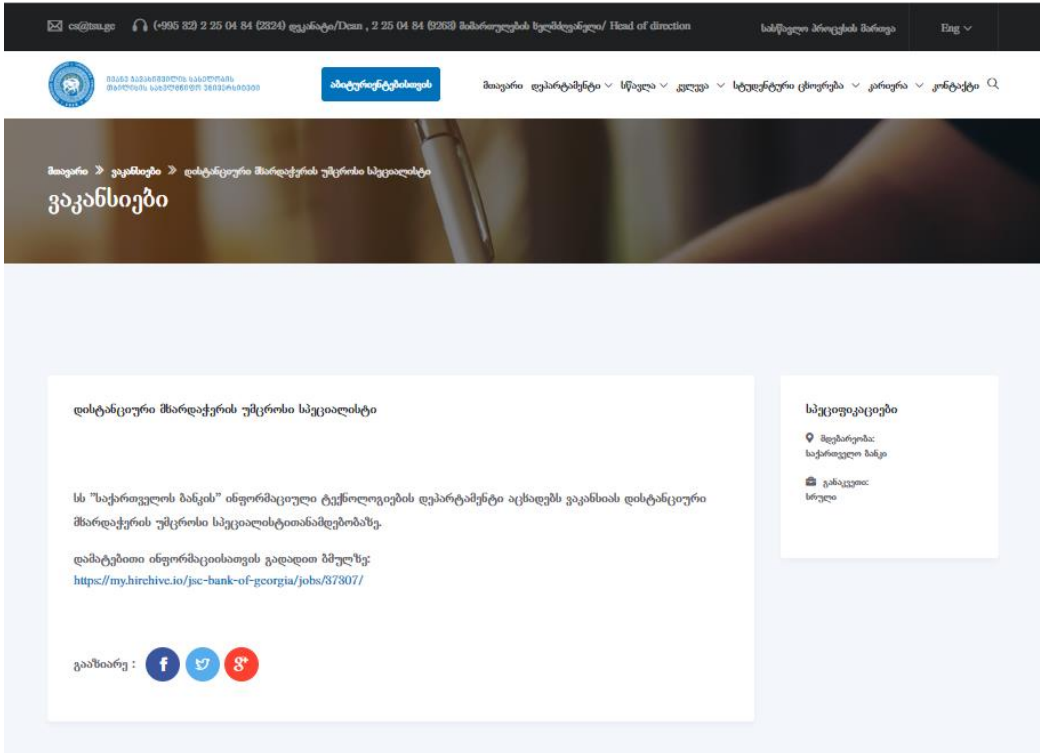
## ვაკანსიები

დისტანციური მწარდაჭერის უმცროსი სპეციალისტი

სრულა


საქართველო ბანკი

დეტალურად



# გაცვლითი პროგრამების გვერდი

✉ [info@tsu.edu.ge](mailto:info@tsu.edu.ge) ☎ (+995 32) 2 25 04 84 (2824) დეკანი/Dean , 2 25 04 84 (8268) შიშროულების ხელმძღვანელი/ Head of direction სასწავლო პროცესის მართვა [Eng](#) ▾

 **თბილისის სახელმწიფო უნივერსიტეტი** [აბიტურენტებისთვის](#) [შიშროუ](#) [დეპარტამენტი](#) ▾ [სწავლა](#) ▾ [კულევა](#) ▾ [სტუდენტური ცხოვრება](#) ▾ [კარიერა](#) ▾ [კონტაქტი](#) 🔍

შიშროუ >> [გაცვლითი პროგრამები](#)

## კარიერა

**9**  
აგვისტო 2018

კონკურსი ზარლანდის უნივერსიტეტის სადოქტორო კვლევითი სტიპენდიების მოსაპოვებლად

[დეტალურად](#)

**9**  
ივნისი 2019

კონკურსი ლუქსემბურგის უნივერსიტეტის სადოქტორო პროგრამაზე

[დეტალურად](#)



# ტექსტური გვერდები



ამიტორიტეტისთვის

მთავარი დეპარტამენტი ▾ სწავლა ▾ კვლევა ▾ სტუდენტური ცხოვრება ▾ კარიერა ▾ კონტაქტი 🔍




## კომპიუტერულ მეცნიერებათა დეპარტამენტი




ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტის კომპიუტერულ მეცნიერებათა დეპარტამენტი შექმნილია გამოყენებითი მათემატიკისა და კომპიუტერულ მეცნიერებათა ფაკულტეტის, რომელიც თსუ-ში დაარსდა 1963 წელს კიბერნეტიკის ფაკულტეტის სახელწოდებით (პირველი მიღება ფაკულტეტს ჰქონდა 1964 წელს). ფაკულტეტის დაარსება განაპირობა ორმა ფაქტორმა: პირველ რიგში ეს იყო ის მიღწევები გამოთვლითი ტექნიკისა და კიბერნეტიკის განვითარებაში, რომელთა გამოყენებამ ადამიანის მოღვაწეობის თითქმის ყველა სფეროში, და განსაკუთრებით საბუნებისმეტყველო დარგებისა და ტექნიკის პრობლემების გადაწყვეტაში ითამაშა მნიშვნელოვანი როლი და მეორე, საქართველოში უკვე არსებობდა კიბერნეტიკის ინსტიტუტი, რომელსაც სჭირდებოდა კადრები. სწორედ ამ ინსტიტუტის დირექტორის, საქართველოს მეცნიერებათა აკადემიის ნამდვილი წევრის, ვლადიმერ ჭავჭავაძის აქტიური მოღვაწეობის შედეგი იყო ახალი ფაკულტეტის გახსნა თსუ-ში, რომელსაც მხარს უჭერდა და უმხარუბოდა უნივერსიტეტის რექტორი, საქართველოს მეცნიერებათა ნამდვილი წევრი ვკვნი სარაძე. ეს იყო ყოფილი საბჭოთა კავშირის მასშტაბით პირველი და უპრეცედენტო შემთხვევა უმაღლეს სასწავლებელში ასეთი ფაკულტეტის გახსნის. ფაკულტეტის პირველი დეკანი


# კონტაქტის გვერდი

 თბილისის სახელმწიფო უნივერსიტეტი [ანტიკორუფციისთვის](#) [შთავარი](#) [ფეხბრტყაშტტი](#) [სწავლა](#) [კვლევა](#) [სტუდენტური ცხოვრება](#) [კარიერა](#) [კონტაქტი](#)

შთავარი > კონტაქტი  
**კონტაქტი**

 (+995 32) 2 25 04 84 (2324)  
დეკანი/Dean , 2 25 04 84  
(9263) შიმართულების  
სელმძღვანელი/ Head of direction

 [cs@tsu.ge](mailto:cs@tsu.ge)

 თბილისი, ი.ჭავჭავაძის გამზ. 3, 0179,  
თსუ, II კორპუსი.

## დაგვიტოვეთ შეტყობინება

სახელი

ელ.ფოსტა

ტელეფონის ნომერი

შეტყობინება

[გაუზიარა](#)

# რეგისტრაციის ფორმის გვერდი



first name

last name

male ▾

personal number

mm/dd/yyyy

citizenship

Adress

email

Phone number

Original high-school graduation diploma

No file chosen

Transcript of academic record

No file chosen

Copy of passport

No file chosen

Internationally accepted English Language Certificate

No file chosen

register

# საიტის სამართავი პანელის ინტერფეისის აწყობა





































## მთავარი გვერდი

The screenshot displays a web management interface. On the left is a dark sidebar with navigation options: გამართვა, admin!, გვერდები, სახელები, რეგისტრირებულები, სლაიდერი, პროგრამები, სტრუქტურა, თანამშრომლები, მრჩეველთა საბჭო, კვლევები, and პროექტები. The main content area is titled 'გვერდები' and contains a section 'ყველა გვერდი'. Below this is a table with 5 rows, each representing an article with an ID, a title, and an edit button.

ID	სათაური	რედაქტირება
1	ჩვენ შესახებ	რედაქტირება
2	სასწავლო პროცესი თსუ-ში	რედაქტირება
3	სასწავლო ცხრილები	რედაქტირება
4	სამეცნიერო წრე	რედაქტირება
5	აბიტურიენტებისთვის	რედაქტირება

# ჩამონათვალის ინტერფეისი

## სიხსლემები


ID	სტატუსი	სათაური	დამატების თარიღი	
17		 "შნგანე ყუთი" უნივერსიტეტში	27-06-19 08:42	 
18		 კომპიუტერული მეცნიერებების მიმართულ...	27-06-19 14:24	 
14		 კომპიუტერული მეცნიერებების მიმართულ...	22-06-19 08:32	 
15		 კომპიუტერული მეცნიერებების მიმართულ...	22-06-19 08:34	 
16		 სტაჟირება საქართველოს ბანკში	22-06-19 08:38	 
12		 კონკურსი ზაარლანდის უნივერსიტეტის ს...	09-06-19 10:55	 
13		 კონკურსი ლუქსემბურგის უნივერსიტეტის...	09-06-19 11:00	 
11		 დისტანციური მხარდაჭერის უმცროსი სპე...	09-06-19 10:22	 
10		 საქართველოს ბანკი იწყებს რიგით მესა...	22-05-19 20:17	 

# დამატების ინტერფეისი

## დამატება

ქარ  Eng

სათაური

აღწერა 

სურათი

დამატების თარიღი

გამოქვეყნებულია

# რედაქტირების ინტერფეისი

## "მწვანე ყუთი" უნივერსიტეტში

ქარ Eng

სათაური "მწვანე ყუთი" უნივერსიტეტში

აღწერა

</> **B** *I* U

სტუდენტური ინიციატივით სახელმწიფო უნივერსიტეტის კომპიუტერული მიმართულების დეპარტამენტი ჩაერთო პროექტ "მწვანე ყუთში"


პროექტის ფარგლებში უნივერსიტეტის XI კორპუსის ტერიტორიაზე განტავსდება "მწვანე ყუთები" და მოხდება მაკულატურის შეგროვება.

შეგროვებულ მაკულატურას უნივერსიტეტი გადასცემს წიგნების მაღაზია [ბიბლუსს](#), სანაცვლოდ კი მიიღებს წიგნებს. აქციის ფარგლებში მიღებული წიგნების გამოყენება მოხდება საქველმოქმედო მიზნით.

მთავარი სურათი

ფაილების ატვირთვა

ატვირთული სურათი



მთავარი სურათი

ფაილების ატვირთვა

არჩევა

ატვირთული სურათი



დამატების  
თარიღი

06/28/2019



გამოქვეყნებულია

განახლება



## ძრავი

სამაგისტრო ნაშრომში ჩატარებული სამუშაო მოიცავდა ბევრ მოდულს. ნაშრომში აღწერილია მხოლოდ ერთი: სიახლეების მოდულის ფუნქციონალის შექმნა

### ლარაველის ინსტალაცია

ვებ-გვერდის ძრავის აწყობა დაიწყო პერსონალურ კომპიუტერში და მხოლოდ გამზადებული ვერსია აიტვირთა სერვერზე. ამისათვის პირველ რიგში სამუშაო კომპიუტერზე დაინსტალირდა ლოკალური სერვერი. უახლესი Laravel-ის მოთხოვნების შესაბამისად სერვერზე დაინსტალირდა 7.1.3 PHP ვერსია და mysql 5.7. მას შემდეგ რაც ლოკალური სერვერი გაიმართა, დაინსტალირდა composer. composer-ის საშუალებით შემდეგი ბრძანებით მოხდა laravel-ის ახალი პროექტის ინსტალაცია:

```
composer create-project laravel/laravel computingTsu.
```

შეიქმნა ბაზა phpmyadmin-ში და დაუკავშირდა შექმნილ პროექტს.

Laravel framework-ისთვის წინასწარ შექმნილია package-ები (პროგრამული პროდუქტი კონკრეტული ამოცანების გადასაჭრელად). სამაგისტრო ნაშრომში გამოყენებულია 3 package:

1. Laravel localization - საიტის მრავალენოვანი სტრუქტურის ასაწყობად.  
<https://github.com/mcamara/laravel-localization>
2. Laravel translation manager - სტატიკური ტექსტების სათარგმნად  
<https://github.com/barryvdh/laravel-translation-manager>
3. Laravel breadcrumbs - განვლილი გზის გასაკეთებლად  
<https://github.com/davejamesmiller/laravel-breadcrumbs>

### ადმინისტრატორის ავტორიზაცია

იმისათვის რომ სამართავ პანელზე წვდომა მიიღოს მხოლოდ წინასწარ შერჩეულმა ადმინისტრატორებმა, ამისთვის გაკეთდა 1 დონიანი ავტორიზაცია. ამისთვის გამოყენებულია

Laravel-ში წინასწარ გამზადებული ავტორიზაციის ფუნქციონალი. ადმინისტრატორების დამატება ხდება სამართავი პანელიდან სუპერ ადმინისტრატორის მიერ, რომელიც ფუნქციონალის გაკეთების დროს შეიქმნა.

ძრავის აწყობა განვიხილოთ სიახლეების მოდულის მიხედვით. საჭირო ფუნქციონალი:

- სიახლის დამატება
- სიახლის რედაქტირება
- სიახლის წაშლა
- სიახლე გამოქვეყნებულია თუ დამალული

სიახლეებისთვის საჭირო ველები:

- სათაური (თარგმნადი)
- აღწერა (თარგმნადი)
- მთავარი სურათი
- დამატების თარიღი
- გამოქვეყნებული იყოს თუ არა

### საიტის სტრუქტურის შესაბამისი ცხრილების შექმნა

ცხრილებისა და მოდელების შექმნა დაიწყო მიგრაციების შექმნით: სიახლეების მოდულისთვის საჭირო იყო ორი ცხრილის შექმნა, თარგმნადი და არათარგმნადი ველებისთვის ცალცალკე:

```
php artisan make:migration create_news_table -m
```

```
php artisan make:migration create_news_details_table -m
```

```

<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateNewsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('news', function (Blueprint $table) {
            $table->increments('id');
            $table->text('image');
            $table->date('date');
            $table->boolean('published');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('news');
    }
}

<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateNewsDetailsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('news_details', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('news_id');
            $table->string('title');
            $table->text('description');
            $table->string('lang');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('news_details');
    }
}

```

php artisan migrate - ბრძანების საშუალებით composer-მა თვითონ შექმნა პროექტის ბაზაში მიგრაციების შესაბამისი ცხრილები და ველები და ასევე laravel-ის მოდელების ჩამონათვალში დაემატა ორი ფაილი: News.php და NewsDetails.php.

```
<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class News extends Model
{
    protected $fillable = [
        'published', 'image', 'date',
    ];

    public function detail()
    {
        return $this->hasMany(NewsDetail::class);
    }
}
```

```
<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class News extends Model
{
    protected $fillable = [
        'published', 'image', 'date',
    ];

    public function detail()
    {
        return $this->hasMany(NewsDetail::class);
    }
}
```

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class NewsDetail extends Model
{
    protected $fillable = [
        'news_id', 'title', 'description', 'lang',
    ];
    public function news()
    {
        return $this->belongsTo(News::class);
    }
}

```

## მართვის პანელის ინტერფეისის ძრავთან დაკავშირება

განისაზღვრა შესაბამისი route-ები web.php ფაილში:

```

// news

Route::group(['prefix' => 'news'], function () {
    Route::get('/', 'AdminController@news');
    Route::get('/create', 'AdminController@createNews');
    Route::post('/create', 'AdminController@storeNews');
    Route::get('/edit/{news}', 'AdminController@editNews');
    Route::post('/edit/{news}', 'AdminController@updateNews');
    Route::get('/delete/{news}', 'AdminController@deleteNews');
    Route::post('/publish', 'AdminController@newsPublish');
});

```

შეიქმნა AdminController შესაბამისი ბრძანებით:

```
php artisan make:controller AdminController
```

AdminController-ში შეიქმნა შესაბამისი ფუნქციები:

ბაზაში არსებული სიახლეების ჩამონათვალის გამოტანა მართვის პანელში:

```
// news

public function news()
{
    $news = News::orderBy('date', 'DESC')->paginate(10);
    return view('administrator.news.items', compact('news'));
}
```

სხალი სიახლის შექმნის ფორმაზე გადასვლა:

```
public function createNews()
{
    return view('administrator.news.create');
}
```

შევსებული ფორმის შენახვა ბაზაში:

```
public function storeNews(Request $request)
{
    $this->validate($request, [
        'image' => 'required',
        'date' => 'required',
    ]);
    foreach (LaravelLocalization::getSupportedLocales() as $localeCode => $properties) {
        $this->validate($request, [
            'title_' . $localeCode => 'required',
            'description_' . $localeCode => 'required',
        ]);
    }
    $image = $request->image[0];
    $extension = $image->getClientOriginalExtension();
    $image_name = '/upload/news/' . str_random() . '.' . $extension;
    $image_path = base_path($image_name);
    file_put_contents($image_path, file_get_contents($image));

    $new = News::create([
        'image' => $image_name,
        'date' => $request->date,
        'published' => $request->published == 'on' ? true : false,
    ]);
    foreach (LaravelLocalization::getSupportedLocales() as $localeCode => $properties) {
        $new->detail()->create([
            'title' => $request->input('title_' . $localeCode),
            'description' => $request->input('description_' . $localeCode),
            'lang' => $localeCode,
        ]);
    }
    return redirect('/admin/news')->with(['success' => 'ინფორმაცია წარმატებით შეიცვალა']);
}
```

არსებული სიახლის რედაქტირების ფორმაზე გადასვლა:

```

public function editNews(News $news)
{
    return view('administrator.news.update', compact('news'));
}

```

სიახლის რედაქტირების ფორმის შენახვა:

```

public function updateNews(Request $request, News $news)
{
    $this->validate($request, [
        'date' => 'required',
    ]);
    foreach (LaravelLocalization::getSupportedLocales() as $localeCode => $properties) {
        $this->validate($request, [
            'title_' . $localeCode => 'required',
            'description_' . $localeCode => 'required',
        ]);
    }
    $image = $request->image[0];
    if (!empty($image)) {
        File::delete(base_path($news->image));
        $extension = $image->getClientOriginalExtension();
        $image_name = '/upload/news/' . str_random() . '.' . $extension;
        $image_path = base_path($image_name);
        file_put_contents($image_path, file_get_contents($image));
    } else {
        $image_name = $news->image;
    }
    $update = $news->update([
        'image' => $image_name,
        'published' => $request->published == 'on' ? true : false,
    ]);
    foreach (LaravelLocalization::getSupportedLocales() as $localeCode => $properties) {
        $news->detail()->where('lang', $localeCode)->first()->update([
            'title' => $request->input('title_' . $localeCode),
            'description' => $request->input('description_' . $localeCode),
        ]);
    }
    if ($update) {
        $request->session()->flash('status', 'success');
        $request->session()->flash('message', 'ინფორმაცია წარმატებით განახლდა');
        return redirect('/admin/news');
    }
    $request->session()->flash('status', 'error');
    $request->session()->flash('message', 'დაფიქსირდა შეცდომა');
    return redirect('/admin/news');
}

```

არსებული სიახლის წაშლა ბაზიდან:

Javascript-ის საშუალებით, ნაგვის ურნაზე დაკლიკებისას ჩნდება მოდული კოთხვით: „თანახმა ხარ თუ არა რომ წაიშალოს ჩანაწერი? “ დათანხმების შემთხვევაში სერვერზე იგზავნება

ბრძანება და მხოლოდ ამის შემდეგ ხდება ბაზიდან ინფორმაციის წაშლა. კონტროლერიდან უკან ბრუნდება პასუხი წაიშალა თუ არა ჩანაწერი და შესაბამისი პასუხი გამოჩნდება ადმინისტრატორის ინტერფეისში.

```
$('.delete').click(function(e) {  
    var target = $(this).attr('href');  
    e.preventDefault();  
  
    $.confirm({  
        title: 'დასტური',  
        content: 'დარწმუნებული ხართ, რომ გურთ ამის სამუდამოდ წაშლა?',  
        buttons: {  
            confirm: {  
                text: 'წაშლა',  
                btnClass: 'btn-red',  
                action: function(){  
                    location.replace(target)  
                }  
            },  
            close: {  
                text: 'დახურვა',  
                action: function(){  
                }  
            }  
        }  
    });  
});
```



```

public function deleteNews(Request $request, News $news)
{
    File::delete(base_path($news->image));
    foreach ($news->detail as $key => $detail) {
        $detail->delete();
    }

    $delete = $news->delete();

    if ($delete) {
        $request->session()->flash('status', 'success');
        $request->session()->flash('message', 'ინფორმაცია წარმატებით წაიშალა');
        return back();
    }
    $request->session()->flash('status', 'error');
    $request->session()->flash('message', 'დაფიქსირდა შეცდომა');
    return back();
}

```

სიახლის გამოქვეყნების მართვა:

სიახლეების ჩამონათვალში არის მოსანიშნი შემდეგი მახასიათებლებით

```



```

Ajax ტექნოლოგიის საშუალებით, მოსანიშნის ცვლილებისას ხდება სერვერზე ბრძანების გაგზავნა და სერვერიდან შესაბამისი პასუხის მიღება: შესრულდა თუ არა ბრძანება, შესაბამისი პასუხის ადმინისტრატორისთვის ინტერფეისში გამოჩენა

```

$('.publish').change(function(event) {
    var id = $(this).attr('id');
    var published = ($(this).is(':checked')) ? 1 : 0;
    $.ajax({
        headers: { 'X-CSRF-TOKEN': $('meta[name="_token"]').attr('content') },
        url: '{{url("admin/news/publish")}}',
        type: "post",
        data: { id: id, published: published, _token: '{{ csrf_token() }}' },
        success: function (response) {
            var res = $.parseJSON(response)
            if (res.status == 'success'){
                new PNotify({
                    text: 'წარმატებით განახლდა',
                    type: 'success',
                    styling: 'bootstrap'
                });
            } else {
                ...
            } else {
                new PNotify({
                    text: 'დაფიქსირდა შეცდომა',
                    type: 'error',
                    styling: 'bootstrap'
                });
            }
        },
        error: function(jqXHR, textStatus, errorThrown) {
            alert(2)
        }
    });
});

```

```

public function newsPublish(Request $request)
{
    $status = array('status' => 'error');
    $video = News::find($request->id);
    $update = $video->update([
        'published' => $request->published,
    ]);
    if ($update) {
        $status['status'] = 'success';
    }
    return json_encode($status);
}

```

ბაზაში არსებული ინფორმაციის გამოტანა საიტის ინტერფეისში

განისაზღვრა შესაბამისი რუტები web.php ფაილში:

```

Route::get('/news', 'PagesController@news')->name('news');
Route::get('/news/{news}', 'PagesController@news_in')->name('news_in');

```

შეიქმნა PagesController შესაბამისი ბრძანებით:

php artisan make:controller PagesController

PagesController შეიქმნა შესაბამისი ფუნქციები:

სიახლეების ჩამონათვალის გამოტანა ვებ-გვერდის მთავარ გვერდზე:

```

public function index()
{
    $today = Carbon::now();
    $sliders = Slider::where('published', true)->orderBy('ordering', 'asc')->get();
    $first_news = News::where('published', true)->orderBy('date', 'desc')->first();
    $news = News::where('published', true)->orderBy('date', 'desc')->limit(4)->get();
    $projects = Project::where('published', true)->where('student', false)->where('date', '>=', $today)->orderBy('date', 'desc')->get();
    return view('client.main.welcome', compact('sliders', 'news', 'projects', 'first_news'));
}

```

სიახლეების ჩამონათვალის გამოტანა ვებ-გვერდზე სიახლეების ჩამონათვალში:

```

public function news()
{
    $news = News::where('published', true)->orderBy('date', 'desc')->paginate(12);
    return view('client.news.main', compact('news'));
}

```

სიახლეების ჩამონათვალის გამოტანა ვებ-გვერდზე სიახლის შიდა გვერდზე:

```

public function news_in(News $news)
{
    $otherNews = News::where('published', true)->where('id', '!=', $news->id)->orderBy('date', 'desc')->get();
    return view('client.news.item', compact('news', 'otherNews'));
}

```

## საიტის ინტერფეისიდან მეილის ფორმის ფუნქციონალის გაკეთება

კონტაქტის ვიუში ფორმის ჩასმა

```

<form action="" method="POST" class="card p-4 p-md-5 shadow-v1">
  <div class="row mt-5 mx-0">
    <div class="col-md-4 mb-4">
      <input type="text" name="name" class="form-control" placeholder="@lang('placeholder.name')" required>
    </div>
    <div class="col-md-4 mb-4">
      <input type="email" name="email" class="form-control" placeholder="@lang('placeholder.email')" required>
    </div>
    <div class="col-md-4 mb-4">
      <input type="tel" name="phone" class="form-control" placeholder="@lang('placeholder.phone')">
    </div>
    <div class="col-12">
      <textarea class="form-control" name="message" placeholder="@lang('placeholder.message')" rows="7"></textarea>
      <button type="submit" class="btn btn-primary mt-4">@lang('site.send')</button>
    </div>
  </div>
</form>
...

```

განისაზღვრა შესაბამისი რუტები web.php ფაილში:

```

Route::get('/contact', 'PagesController@contact')->name('contact');
Route::post('/contact', 'PagesController@sendEmail');

```

PagesController-ში შეიქმნა შესაბამისი ფუნქცია:

```

public function contact()
{
    $contact = Contact::first();
    return view('client.contact.main', compact('contact'));
}

public function sendEmail(Request $request)
{
    $contact = Contact::first()->email;
    $this->validate($request, [
        'name' => 'required|max:50',
        'phone' => 'required',
        'email' => 'required|max:50',
        'message' => 'required|max:300',
    ]);
    $data = array(
        'email' => $request->email,
        'name' => $request->name,
        'phone' => $request->phone,
        'bodyText' => $request->message,
    );
    try {
        Mail::send('emails.contactSend', $data, function ($message) use ($data, $contact) {
            $message->from($data['email']);
            $message->to($contact);
            $message->subject($data['name'] . '-' . $data['phone']);
        });
        return back()->with("success", "თქვენი შეტყობინება წარმატებით გაიგზავნა");
    } catch (\Swift_TransportException $e) {
        return back()->with("error", "თქვენი შეტყობინება ვერ გაიგზავნა, სცადეთ ახლიდან!");
    }
}

```

ელ.ფოსტის გასაგზავნად გამოვიყენეთ SMTP ავტორიზაცია, რომელსაც სჭირდება მეილ სერვერი რომელიც დაყენებულია ჰოსტინგზე. შეიქმნა მეილ ექაუნთი no-replay@computing.tsu.ge. იმისთვის, რომ მეილების გაგზავნა მოხდეს სწორად და თავიდან ავიცილოთ spam-ში მოსვლა სწორედ ამიტომ კეთდება SMTP ავტორიზაცია, ამ დროს ექაუნთი გადის ავტორიზაციას და გაგზავნაც ავტორიზებული ექაუნთით ხდება.

## ვებ-გვერდის სერვერზე გამართვა

როდესაც Laravel პროექტს ლოკალურად კომპიუტერში ვაყენებთ ფაილური სტრუქტურა არის ისეთი, რომ მთავარი გამშვები ფაილი, რომელსაც ყველა სერვერი კონფიგურაციის მიხედვით უშვებს პირველად (index.php) მოთავსებულია არა მთავარ დირექტორიაში არამედ public საქაღალდეში, ამიტომ როდესაც პროექტის დასტარტვა ხდება ლოკალურად ეშვება ბრძანება php artisan serve და ამ ბრძანების მიხედვით <http://localhost:8000> ისტარტება სერვერი და ეს სერვერი უყურებს პროექტში მთავარ დირექტორიაში public საქაღალდეში არსებულ index.php-ს. იმავენაირად ხდება სერვერზე გამართვაც. თუ ვიყენებთ ვირტუალურ სერვერს (VPS), შეგვიძლია იგი დავაკომფიგურიროთ ისე, რომ სერვერი უყურებდეს არა მთავარ დირექტორიას არამედ მთავარ დირექტორიაში არსებულ public საქაღალდეს და ვებ-გვერდზე შესვლის დროს ყველაფერი იმუშავებს კარგად, თუმცა მსგავსი რამ ვერ ხერხდება საზიარო ჰოსტინგზე. ამ შემთხვევაში რა კონტენტიც არის public საქაღალდეში, უნდა ამოვიდეს ერთი ნაბიჯით ზემოთ, რაც გვადლევს იმ შედეგს რომ მთავარი გამშვები ფაილი (index.php) უკვე იქნება მთავარ დირექტორიაში და აქაც უპრობლემოდ გაეშვება პროექტი. მოცემულ შემთხვევაში გვექონდა საზიარო ჰოსტინგი და შესაბამისად განხორციელდა კონფიგურაცია.

## დასკვნა

სამაგისტრო ნაშრომის შედეგად შექმნილია ვებ-გვერდი რომელიც განთავსებულია მისამართზე: <https://computing.tsu.ge> და მისი სამართავი პანელი: <https://computing.tsu.ge/admin>. ვებ-გვერდზე ასახულია ყველა ის ინფორმაცია, რისი განთავსებაც მიზნად იქნა დასახული ნაშრომის შექმნისას: დეპარტამენტის შესახებ სრულყოფილი ინფორმაცია, სასწავლო პროგრამების შესახებ ინფორმაცია, დეპარტამენტში მიმდინარე კვლევები და პროექტის შესახებ ინფორმაცია, სტუდენტური ცხოვრება, საკონტაქტო ინფორმაცია, სიახლეები, სასწავლო ცხრილები და.შ ასევე ვებ-გვერდის საშუალებით შესაძლებელია უკუკავშირის დამყარება მომხმარებელთან კონტაქტის გვერდიდან მეილის გაგზავნის სახით. უცხო ქვეყნის მოქალაქეებისთვის შექმნილია თსუ-ში ჩარიცხვისათვის შესავსები ონლაინ აპლიკაციის ფორმა.

დასახული მიზნის მისაღწევად, სამაგისტრო ნაშრომის შექმნისას გამოყენებულია შემდეგი ვებ ტექნოლოგიები:

ინტერფეისის შექმნისთვის:

- html
- css
- javascript
- javascript ბიბლიოთეკები
- ajax

ძრავის შექმნისთვის: php framework - Laravel

ვებ-გვერდის ტესტირებამ აჩვენა, რომ ვებ-გვერდის ფუნქციონალი გამართულად მუშაობს და მოხდა მისი რეალური ინფორმაციით შევსება. შედეგად ვებ-გვერდი ხელმისაწვდომია სტუდენტებისთვის, ლექტორებისა და ნებისმიერი სხვა დაინტერესებული პირისთვის.

სამაგისტრო ნაშრომზე მუშაობისას გამოიკვეთა, რომ შექმნილი პროდუქტი საშუალებას იძლევა კიდევ უფრო განვითარდეს ვებ-გვერდი, ამ სისტემაშივე მოხდეს კათედრაზე მიმდინარე პროცესების სისტემიზაცია.



## ლიტერატურა

1. Andreas Mauthe; Peter Thomas (2004). *Professional Content Management Systems: Handling Digital Media Assets*. John Wiley & Sons. ISBN 978-0-470-85542-3.
2. "Database Structure and Manipulation Capabilities of a Picture Database Management System (PICDMS) - IEEE Xplore Document". [ieeexplore.ieee.org](http://ieeexplore.ieee.org). Retrieved February 18, 2017.
3. Maks Surguy (July 27, 2013). "History of Laravel PHP framework, Eloquence emerging". [maxoffsky.com](http://maxoffsky.com). Retrieved May 10, 2015.
4. "Release - laravel/framework". [github.com](https://github.com). Retrieved July 20, 2019.
5. "Laravel documentation (version 4.0): Query builder". [laravel.com](http://laravel.com). Retrieved April 27, 2015.
6. "Laravel Homestead - Laravel the PHP Framework for Web artisans". Retrieved October 30, 2016.
7. Lerdorf, Rasmus (2007-04-26). "PHP on Hormones – history of PHP presentation by Rasmus Lerdorf given at the MySQL Conference in Santa Clara, California". The Conversations Network. Retrieved 2009-12-11.
8. "PHP: News Archive - 2019". 11 July 2019. PHP 7.4.0 alpha 3 Released. Retrieved 11 July 2019.
9. "PHP: Function arguments - Manual". [secure.php.net](http://secure.php.net).
10. "History of PHP". [php.net](http://php.net).
11. "MySQL 8.0 Release Notes". [mysql.com](http://mysql.com). Retrieved 27 April 2019.
12. "Supported Platforms: MySQL Database". Oracle. Retrieved 24 March 2014.
13. "What is MySQL?". *MySQL 5.1 Reference Manual*. Oracle. Retrieved 17 September 2012. The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”)
14. "History of MySQL". *MySQL 5.1 Reference Manual*. MySQL. Retrieved 26 August 2011.
15. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>