



ივანე ჯავახიშვილის სახელობის თბილისის
უნივერსიტეტი

სახელმწიფო

ჯგუფური პროექტი

რეალურ დროში სასაუბრო პროგრამა – CH10

Realtime Chat Application – CH10

საბაკალავრო პროგრამა - კომპიუტერული მეცნიერება

შემსრულებლები:

ირაკლი სართანია,

შალვა ქუმაშვილი,

თენგიზ ჩიგოგიძე,

გიორგი ჯოხაძე.

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი,

კომპიუტერული მეცნიერების დეპარტამენტი

ხელმძღვანელი: ფიზ.მათ. მეცნიერებათა კანდ. დავით ხაჩიძე

თბილისი

2021

სარჩევი

ანოტაცია	3
შესავალი:	4
სასაუბრო (ჩათის) პროგრამული სისტემების ძირითადი არქიტექტურა	5
პროგრამული სისტემა CH10 - დაგეგმვა და განხორციელება	7
პროექტის რეალიზაციის თეორიული და ტექნოლოგიური მეთოდები:	8
ამოცანის ტექნიკური კომპონენტები.....	9
პროგრამული სისტემის პრაქტიკული განხორციელება	11
დასკვნა	19

ანოტაცია

ნაშრომში წარმოდგენილია პროგრამული სისტემის დაგეგმვა და განხორციელება, რომელიც უზრუნველყოფს რეალურ დროში ურთიერთკავშირს მომხმარებელთა შორის. სისტემის შემუშავებაში გათვალისწინებულია ის თავისებურებები, რომელიც დაკავშირებულია საქართველოს რეალობასთან მომხმარებელთა ტექნიკური და ცოდნითი შესაძლებლობების რესურსებით. პროგრამული სისტემის (CH10) კონცეპტუალურ გეგმაში აღწერილია მისი ფუნქციონირების ძირითადი მოდულები და ბიზნეს წესები; შეფასებულია თითოეული მოდულის რეალიზაციისათვის გამოყენებული პროგრამული ინსტრუმენტები. საბოლოო პროდუქტი CH10 არის სრულად ფუნქციონირებადი - საკომუნიკაციო აპლიკაცია, რომელიც მომხმარებლებს აძლევს საშუალებას ერთმანეთთან იკონტაქტონ რეალურ დროში.

Realtime Chat Application – CH10

The paper presents the planning and realization of a software system that provides real time communication between users. The development of the system considers the features related to the reality of Georgia with the resources of technical and knowledge capabilities of consumers. The conceptual plan of the software system (CH10) describes the main modules and business rules of its operation; The software tools used to implement each module are evaluated. The final product of CH10 is a fully functional communication app that allows users to communicate with each other in real time.

შესავალი:

ონლაინ სასაუბრო (გავრცელებული როგორც ტერმინი „ჩატი“) ეს არის რეალურ დროში კომპიუტერულ ქსელში შეტყობინების გაცვლის საშუალება. მისთვის დამახასიათებელი ნიშანია რეალურ დროში კომუნიკაცია. ეს გამოარჩევს ჩათს სხვადასხვა კონტაქტის საშუალებებისგან, მაგალითად ფორუმებისგან. თუ ფორუმებში შეიძლება დაიწეროს კითხვა და დაელოდო პასუხს (ამავდროულად შეიძლება რამოდენიმე სხვა მომხმარებლისგან მივიღოთ პასუხი), ჩათში ხდება ურთიერთობა მხოლოდ მათთან, ვინც მოცემულ მომენტში იმყოფება ადგილზე და ურთიერთობის შედეგები, შესაძლოა, არც კი იქნას შენახული. სიტყვა „ჩათში“ იგულისხმება ჯგუფური ურთიერთობა, თუმცა შესაძლებელია ისიც, რომ ურთიერთობა მიმდინარეობდეს ერთი ერთზე.

დღესდღეობით, ონლაინ ჩათები მრავალი ადამიანის ცხოვრების ერთ-ერთი ნაწილია. ის შეიძლება გამოვიყენოთ მეგობრებთან სალაპარაკოდ, ან ბიზნესის მიზნით. Covid-19-ის გამო ასეთი სახის აპლიკაციების გამოყენება უფრო გახშირდა. ჩათები ბევრგან გვხვდება : ვეისბუკში, ონლაინ თამაშებში და ა.შ.

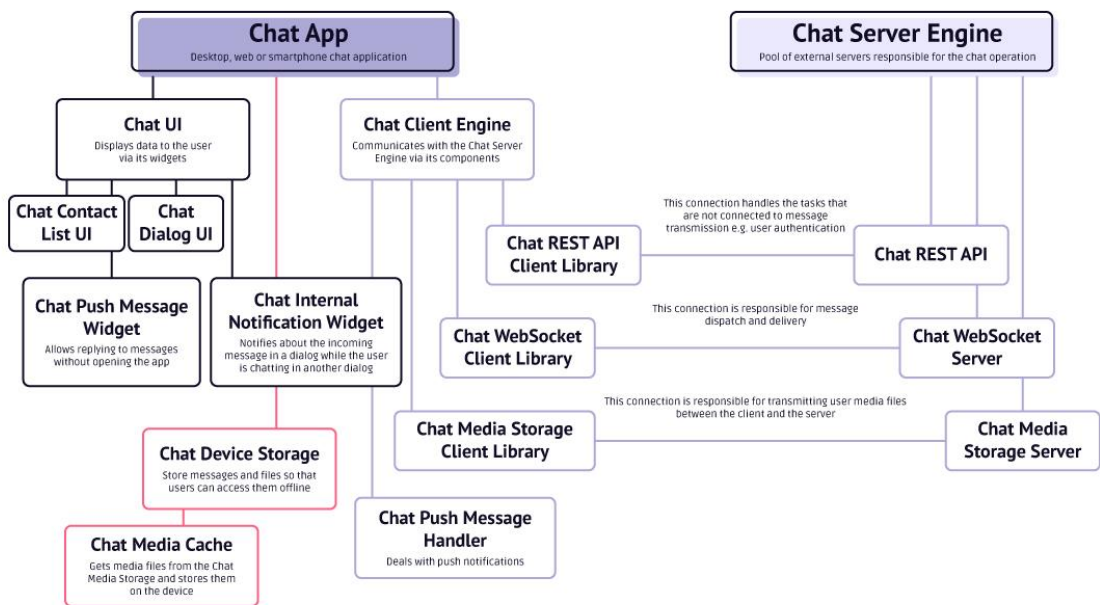
როგორც ავლინებით, ონლაინ ჩათების მიზანი არის მესიჯების გაგზავნა გამგზავნისა და მიმღებს შორის. ზოგიერთი ჩათის მეშვეობით შეიძლება სურათების, ვიდეოსა და ხმის ჩანაწერის გაგზავნაც. რამოდენიმე ონლაინ ჩათის აპლიკაცია მოითხოვს მომხმარებლის დარეგისტრირებას იმის მიზნით, რომ მოხდეს მეტსახელის (nickname) აუტენტიკაცია, რომლის გამოყენებაც მომხმარებელს სურს. მაგალითად, დისკორდის ან სკაიპის გამოსაყენებლად მომხმარებელმა უნდა გაიაროს ავტორიზაცია დარეგისტრირებული აქაუნთით. ჩათები შეიძლება იყოს browser based ან client based. Browser based სახის ჩათებში, მომხმარებელი იყენებს ონლაინ ჩათის ინტეფეისს ბრაუზერშივე, ამის მაგალითი არის ტვიტერი. Client based ჩათის შემთხვევაში, მომხმარებელმა უნდა დააინსტალიროს ონლაინ ჩათის აპლიკაცია თავიანთ კომპიუტერზე, ასეთი მაგალითია სკაიპი. დღეს უმეტეს ჩათის აპლიკაციას აქვს ორივეს შესაშლებლობა, მაგალითად დისკორდს. ასევე არსებობს ისეთი ჩათებიც, სადაც შეიძლება VoIP სერვისის გამოყენება (ამაში იგულისხმება დარეკვის საშუალება). ასეთი ფუნქციონალი ბევრ ჩათის აპლიკაციას აქვს, მაგალითად დისკორდს და სკაიპს, ასევე აქვთ კამერების მხარდაჭერაც და ფაილების გაგზავნის/მიღების საშუალებაც.

საქართველოს სამომხმარებლო ბაზრის ერთი ნაწილის შესწავლამ, რომელიც ძირითადად წარმოადგენდა სტუდენტები და პედაგოგები აჩვენა, რომ მესიჯების, სურათებისა და ფაილების გაგზავნის, ასევე ღია და დახურული ჯგუფის შექმნის

ფუნქციონალის მქონე სასაუბრო პროგრამა გაუადვილებდა ურთიერთობებს სასწავლო, ასევე საკომუნიკაციო კუთხით. აქედან გამომდინარე საქართველოს რეალობასთან მორგებული (მომხმარებელთა ტექნიკური და ცოდნითი შესაძლებლობების რესურსების გათვალისწინები) პროგრამული ხელსაწყო დაეხმარება მომხმარებელს მისთვის სასურველი კომუნიკაციის უზრუნველყოფაში.

სასაუბრო (ჩათის) პროგრამული სისტემების ძირითადი არქიტექტურა

ზოგადად, ჩათის აპლიკაცია შედგება რამოდენიმე კომპონენტისაგან: შეტყობინებების აპლიკაცია, სერვერი და მუდმივი კავშირი.



შეტყობინებების აპლიკაცია (Client side / Chat App) - ის რასაც მომხმარებელი ხედავს, ეს არის ის სისტემის ნაწილი რომელიც მდებარეობს ტელეფონზე, ლეპტოპზე ან პერსონალურ კომპიუტერზე. ამ მოდულის ასაწყობად გამოიყენება HTML, CSS, JavaScript, AJAX, VBScript.

სერვერი (Server side / Server App) - აპლიკაციის გამოყენებაზე, აპლიკაცია უკავშირდება ცენტრალურ სერვერს. მხოლოდ ამ კავშირის გამო შეგვიძლია გავაგზავნოთ შეტყობინებები სხვებთან, რომლებიც დაკავშირებულნი არიან იმავე სერვერთან. სერვერულ მხარეზე არის სერვერული პროგრამა, რომელშიც არის კლიენტის კავშირის სია.

სერვერი ინარჩუნებს ინფორმაციას, თუ რომელი კავშირი რომელ მომხმარებლებს ეკუთვნის, ასე რომ შეტყობინებები საიმედოდ გადაეცემა სწორ მიმღებს და აღინიშნება, რომ ისინი გაგზავნილია გარკვეული გამგზავნის მიერ. ზუსტად ასე ხდება მონაცემების მიმოცვლა მოდულებს შორის.

სერვერული ნაწილის ასაწყობად ზოგადად გამოიყენება Java, C#, C++, ASP.NET MVC, JSP, Servlet და სხვა. ასევე საჭიროა მონაცემთა ბაზა, სადაც შევინახავთ და წამოვიღებთ მონაცემებს ფუნქციონალის სრულყოფილებისათვის. ძირითადად გამოიყენება SQL Server, MongoDB, Oracle, Firebase და სხვა.

მუდმივი კავშირი - მუდმივი კავშირი. ეს ნიშნავს იმას, რომ აპლიკაცია მუდმივად უნდა იყოს დაკავშირებული სერვერთან. მომხმარებელი იმ შემთხვევაში მიიღებს შეტყობინებას, თუ ის დაკავშირებულია სერვერთან. შეტყობინებას მიიღებს მანამ, სანამ კავშირი სტაბილურია.

ჩათის აპლიკაციებს აქვთ თავისი დადებითი და უარყოფითი მხარე:

უპირატესობები - ონლაინ ჩათებს აქვს ბევრი უპირატესობა, მაგალითად: შეგვიძლია ახალი მეგობრების შეძენა, შევხვდეთ ისეთ ხალხს, რომელსაც აქვთ საერთო ინტერესი, შეიძლება სამყაროს ნებისმიერი კუთხიდან კონტაქტი. ასევე ინტერნეტში საუბარს შეუძლია თავდაჯერების გაზრდა. ბიზნესის მიზნით ჩათის აპლიკაციებს ასეთი უპირატესობები აქვს:

1. სწრაფი კომუნიკაცია: მარტივი არის ლაივ რეჟიმში მოსწავლეებთან, კოლეგებთან ან კლიენტებთან საუბარი. საერთოდ, ლაივ რეჟიმის ჩათის გამოყენებისას მომხმარებელთა პრობლემის მოგვარებას ნაკლები დრო სჭირდება.
2. ნაკლები დანახარჯი: ლაივ ჩათი უფრო იაფია, ვიდრე ტრადიციული სატელეფონო ზარი. მაგალითად, როდესაც მომხმარებელს ექნება რაიმე კითხვა კომპანიასთან, დარეკვის ნაცვლად გამოიყენებს ლაივ ჩათს.
3. რიგის მოლოდინის გარეშე: თუ ჩათის აპლიკაციას ბიზნესის მიზნით ვიყენებთ, მომხმარებელს შეეძლება დაუყოვნებლივ მიმართოს კომპანიის აგენტს ლოდინის გარეშე.
4. ფაილების გაგზავნა: ჩათის მეშვეობით შეიძლება ფაილების სწრაფი გაგზავნა. იმის მაგივრად რომ გამოვიყენოთ იმეილი, ფაილების გაგზავნისთვის შეგვიძლია ჩათის აპლიკაცია გამოვიყენოთ.

ნაკლოვანება - ბევრ ჩათის აპლიკაციას აქვს თავისი ნაკლიც, მაგალითად: ინტერნეტი:

1. მომხმარებელს ჭირდება კარგი ინტერნეტ კავშირი იმისათვის, რომ გამოიყენოს ჩათის აპლიკაცია.
2. კომუნიკაციის სირთულები: ზოგიერთ მომხმარებელს უჭირს ჩათის აპლიკაციების გამოყენება;
3. ვირუსის რისკი: ფაილების გაგზავნის დროს შეიძლება ფაილს მოყვეს ვირუსი.
4. User Unfriendly: ჩათის აპლიკაციის გამოყენება დამწყები კომპიუტერის მომხმარებლისთვის არის რთული.

პროგრამული სისტემა CH10 - დაგეგმვა და განხორციელება

დღესდღეობით არ არსებობს ქართული ონლაინ ჩათები, თუ არსებობს ვერ უძლებს კონკურენციას ისეთი საკომუნიკაციო საშუალებებს როგორცაა ფეისბუქი, სლექი, ინსტაგრამი. პროექტის მიზანია შეიქმნას პირველი ქართული ჩათი, რომელიც რეალურ დროში მისცემს მომხმარებლებს საშუალებას ერთმანეთში განახორციელონ ინფორმაციის მიმოცვლა. CH10-ში მომხმარებლებს ექნებათ იმის საშუალება რომ ერთმანეთს გაუგზავნონ წერილები, ფოტოები, ემოჯი, გიფები და ფაილები. ამავდროულად, მომხმარებელს ექნება ე.წ. user card, რომელიც შეიცავს მცირე ინფორმაციას მომხმარებელზე. ასევე, მომხმარებელს შეუძლია როგორც პირადი ასევე ჯგუფური ჩათები შექმნას, მონიშნოს მისი ფავორიტი ჩათი და ა.შ.

პროექტის ტექნიკური მხარე დაყოფილია სამი ნაწილისაგან:

- Front-end(react)
- Back-end(redux)
- Database(firebase)

პროგრამის განმახორციელებელი გუნდი შედგება 4 წევრისგან წევრების მიერ შესრულებულ სამუშაოებს.

შალვა ქუმაშვილი - back-end დეველოპერი (redux, firebase), შეიმუშავა ფუნქციონალის ლოგიკა, მუშაობდა firebase-თან და რედაქსის საშუალებით აკავშირებდა ფრონტს ბექთან.

ირაკლი სართანია - front-end დეველოპერი (react), მუშაობდა ფრონტის ნაწილზე, გიფების გაგზავნაზე და ფაილების ატვირთვაზე.

გიორგი ჯოხაძე - front-end დეველოპერი (react, tester) მუშაობდა ფრონტის ნაწილზე (ჩენილები, პირადი მიმოწერები), აქტიურად ტესტავდა პროექტს და აღმოაჩინა ხარვეზები, რომელიც შემდგომ ერთად გავასწორეთ.

თენგიზ ჩიგოგიძე - front-end დეველოპერი (react), მუშაობდა ჩენილებსა და პირადი მიმოწერების შექმნაში, იძიებდა სხვადასხვა ჩათებზე ინფორმაციას და გვაწვდიდა ჩვენ რათა საინტერესო ფუნქციონალებით გაგვემდიდრებინა აპლიკაცია.

პროექტის რეალიზაციის თეორიული და ტექნოლოგიური მეთოდები:

ჩათის აპლიკაციის შექმნისას პროგრამისტების ჯგუფმა პირველ რიგში უნდა გადაწყვიტოს ის, თუ რომელ მეთოდოლოგიას გამოიყენებენ პროექტისთვის. განვიხილოთ რამდენიმე სახის მეთოდოლოგია, რომლის მეშვეობითაც შეიძლება ჩათის აპლიკაციის შექმნა, მაგალითად: Waterfall, Agile, Scrum, DevOps, Lean Software Development, Extreme Programming და სხვა.

რაც შეეხება ტექნოლოგიებს, ჩათის აპლიკაციის შექმნისთვის საჭირო არის სერვერი, სადაც შეინახება მესიჯები/შეტყობინებები, კლიენტი სადაც გავაგზავნით და მივიღებთ მესიჯებს, და ისეთი ტექნოლოგია, რომლის მეშვეობითაც შეიძლება ტრანსპორტაცია სერვერსა და კლიენტს შორის. ასევე შეიძლება რაიმე ბიბლიოთეკების და API-ების გამოყენებაც.

ჩათის აპლიკაციის შესაქმნელად გამოიყენება რამდენიმე ხელსაწყო ერთდროულად, რაც აუცილებლად მოითხოვს ერთმანეთთან თავსებადობის შეფასებას. მაგალითად, თუ პროექტი იქმნება ASP.NET MVC აპლიკაციის მეშვეობით, მაშინ უნდა

გამოვიყენოთ SignalR. ამ ბიბლიოთეკის მეშვეობით სერვერულ კოდს შეუძლია გაგზავნოს ასინქრონული შეტყობინებები client-side ვებ აპლიკაციაში. შეიძლება ობიექტზე ორიენტირებული პროგრამული ენების გამოყენებაც, როგორცაა ჯავა, სადაც შეიძლება სერვერის და კლიენტის იმპლემენტაცია. ასევე შეიძლება პითონის გამოყენება, ან რეაქტის/რედაქსის. მონაცემთა ბაზისთვის შეიძლება SQL, MongoDB, FireBase ან სხვა სერვერები. კლიენტსა და სერვერის დამაკავშირებელ ტექნოლოგიად შეიძლება REST API ან სხვა ტექნოლოგიის გამოყენება.

ამოცანის ტექნიკური კომპონენტები

თუ გადავხედავთ პოპულარულ ჩათებს (ფეისბუქი, ინსტაგრამი, დისკორდი), ყველა გამოირჩევა იმითი რომ მათი ინტერფეისი არის მარტივი რაც იწვევს იმას რომ მომხმარებელს არ უჩნდება ზედმეტი შეკითხვა და მისი გამოტენებაც შესაბამისად მარტივია. ამის გამო ჩვენმა გუნდმა გადაწყვიტა რომ დეველოპერებიდან რაღაც რესურსი დაეთმო UI/UX ის კვლევისთვის რათა სხვა ჩათების მაგალითზე შეგვექმნა მარტივი და სწრაფად გამოყენებადი ინტერფეისი. ჩვენ ჩავატარეთ ტესტები სხვასხვა ასაკობრივ ჯგუფებზე. ამან იმ შედეგამდე მიგვიყვანა რასაც ახლა თქვენ ხედავ CH10-ის სახით.

ჩვენი მიზანი იყო არა მხოლოდ თვალისათვის კარგი და მარტივად აღსაქმელი აპლიკაცია შეგვექმნა, არამედ შეგვექმნა მომხმარებლისთვის საინტერესო სივრცე, სადაც ერთმანეთთან დაკონტაქტებას შეძლებდნენ. აქედან გამომდინარე, ჩვენ ვფიქრობდით სხვადასხვა ფუნქციონალებზე, განვიხილავდით მათ და შესაბამისად ვაფასებდით თუ რამდენად ღირებული და მოსწრებადი იქნებოდა ამ ფუნქციონალების დამატება ჩვენს პროექტში.

ახლა ვისაუბროთ უშუალოდ პროექტზე და მასზე ჩატარებულ სამუშაოზე, პროექტი როგორც ზემოთ ავღნიშნეთ შედგება სამი ნაწილისგან:

Front-end

პროექტის ინტეგრაციისთვის ჩვენ გამოვიყენეთ facebook-ის შექმნილი library სახელად React. ამაზე არჩევანი იმის გამო შევაჩერეთ რომ იგი დღეს არის ყველაზე პოპულარული და შესაბამისად ბრაუზერების ძალიან დიდ ნაწილს აქვს მისი საპორტი.

რეაქტი მოიცავს - JSX, CSS & JavaScript.

JSX - ეს არის სინტაქსის გაფართოება JavaScript-ზე. JSX აწარმოებს React ელემენტებს და მისი გამოყენება React-თან არის მიზანშეწონილი იმისათვის, რომ მარტივად ავაგოთ UI. JSX შეიძლება გამოიყურებოდეს როგორც შაბლონური ენა, მაგრამ მას გააჩნია JavaScript-ის სრული ძალა.

CSS3 - კასკადური სტილების ენა. მისი გამოყენებით შეგვიძლია ვაკონტროლოთ ტექსტის ფერი, შრიფტების სტილი, ელემენტების სტილი, ფერები, დიზაინი და ასევე საიტის ვიზუალი სხვადასხვა მოწყობილობებზე.

JavaScript - ეს არის პროგრამირების ენა, რომელიც საშუალებას გვაძლევს განვახორციელოთ რთული ფუნქციები. ჯავასკრიპტი საიტს ხდის დინამიურს. მისი დახმარებით შეგვიძლია გვერდი ხდებდეს განახლებადი დროის მოცემულ მომენტში. არსებობს ენის ბევრი სტანდარტი. ჩვენი საიტი დაეყრდნობა ES6 სტანდარტს და გამოყენებული იქნება უაზლესი ფუნქციონალი. ასევე გამოვიყენებთ ჯავასკრიპტის ერთ-ერთ ბიბლიოთეკას JQuery, რომელიც დახმარებითაც დროის მოკლე დროში დაიწერება კოდი

Back-end & Database

რაც შეეხება პროექტის back-end ნაწილს, ჩვენი მიზანი იყო საიტისთვის შეგვექმნა ისეთი ფუნქციონალი როგორცაა რეგისტრაცია/ავტორიზაცია, როუტინგი, მესიჯების გაგზავნა, მონაცემების(მესიჯები, გაგზავნილი ფაილები, მომხმარებლები) შენახვა მონაცემთა ბაზაში. ამ ფუნქციონალის უზრუნველსაყოფად ჩვენ გამოვიყენეთ Redux.

Redux - ეს არის ჯავასკრიპტის ბიბლიოთეკა, რომლითაც შეგვიძლია აპლიკაციის მდგომარეობის მართვა. Redux-ის საშუალებით შეგვიძლია დავწეროთ აპლიკაციები, რომლებიც იქცევიან თანმიმდევრულად, მუშაობენ სხვადასხვა გარემოში (client, server, და

native). Redux მუშაობს ნებისმიერ UI-სთან და აქვს დანამატების დიდი ეკოსისტემა, რომელიც ჩვენს საჭიროებებს შეესაბამება.

მონაცემების შენახვისთვის გამოვიყენეთ Firebase - ეს არის Google-ის მიერ მხარდაჭერილი პროგრამის შემუშავების პროგრამა, რომელიც დეველოპერებს საშუალებას აძლევს შექმნას iOS, Android და ვებ აპლიკაციები. Firebase გვთავაზობს უამრავ მომსახურებას, მათ შორის:

Analytics - Google Analytics for Firebase გთავაზობთ უფასო, შეუზღუდავ რეპორტაჟს 500 - მდე ცალკეული event-ის შესახებ. Analytics წარმოგიდგინებთ მონაცემებს მომხმარებლის ქცევის შესახებ iOS და Android აპებში, რაც უკეთეს გადაწყვეტილებებს მიიღებს მუშაობის გაუმჯობესებისა და აპების მარკეტინგის შესახებ.

ავთენტიფიკაცია - Firebase ავთენტიფიკაცია დეველოპერებს უმარტივეს ავტორიზაციის უსაფრთხო სისტემების შექმნას და აუმჯობესებს მომხმარებლებისთვის სისტემაში შესვლას. ეს ფუნქცია გვთავაზობს identity solution-ს, ელ.ფოსტისა და პაროლის ანგარიშების მხარდაჭერას, ასევე Google- ს, Facebook- ს, GitHub- ს, Twitter- ის შესვლას და ა.შ.

Realtime მონაცემთა ბაზა - Firebase Realtime მონაცემთა ბაზა წარმოადგენს ღრუბლოვანი NoSQL მონაცემთა ბაზას, რომელიც საშუალებას იძლევა მონაცემთა შენახვა და სინქრონიზაცია მოხდეს მომხმარებლებს შორის რეალურ დროში. მონაცემები სინქრონიზებულია ყველა კლიენტზე რეალურ დროში და კვლავ ხელმისაწვდომია, როდესაც აპი ხაზგარეშე რეჟიმში მიდის.

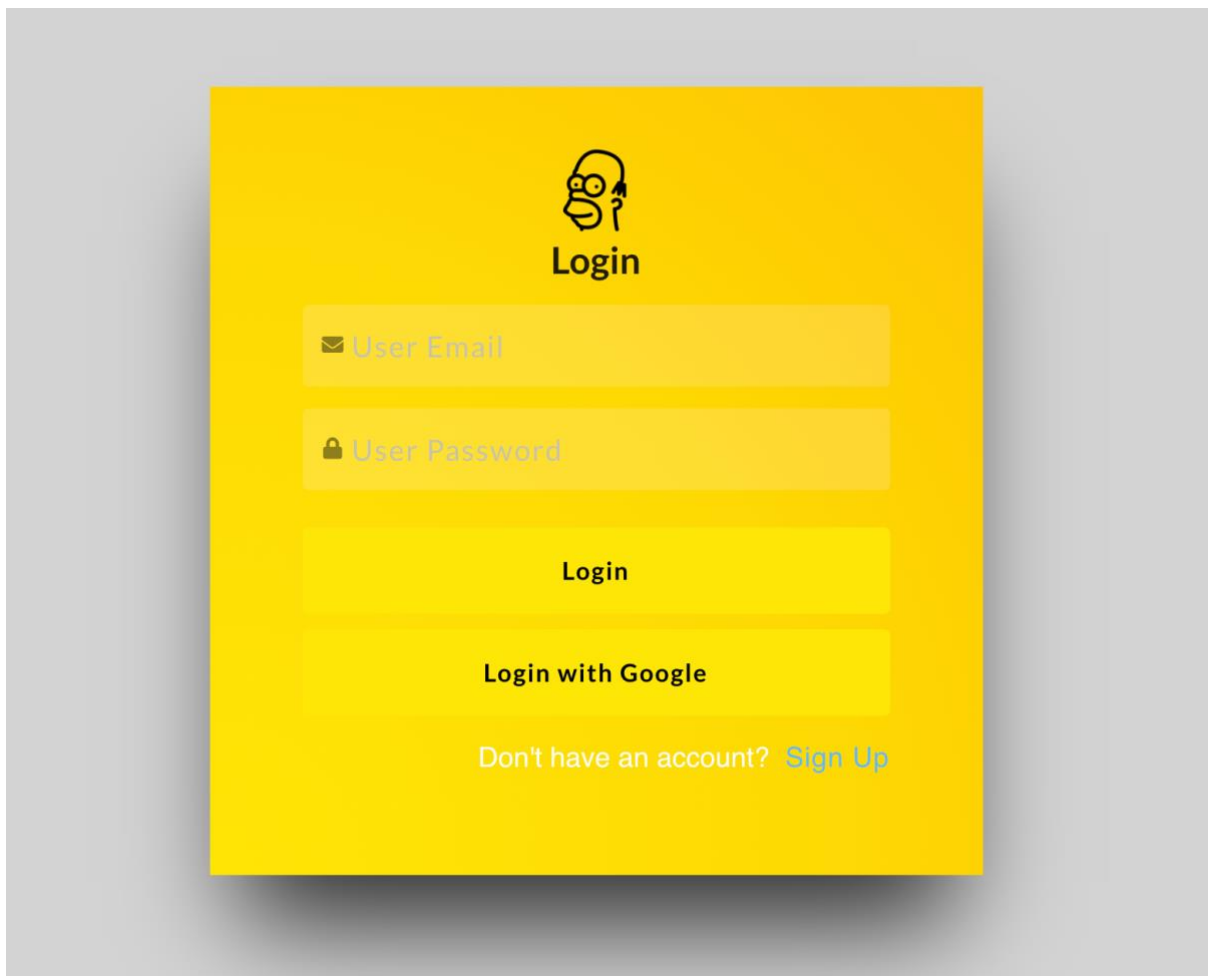
Firebase Cloud Storage - ასევე ცნობილია, როგორც Firebase File Storage, არის ობიექტის შენახვის სერვისი, რომელიც არის Google Cloud Platform- ზე. როდესაც Google Cloud Storage არის ჩართული Firebase აპებში, ჩვენ გვექნება საშუალება აპლიკაციაში ნებისმიერი გაფართოების ფაილის ატვირთვა ან ჩამოტვირთვა .SDK- ის საშუალებით ასევე შეგვიძლია ვმართოთ ჩვენი მედია და მასზე წვდომა შეგვიძლია პირდაპირ ჩვენი მეხსიერების ანგარიშიდან. ინტეგრაცია მხარდაჭერილია Android, C ++, iOS, Unity და ვებ აპებისთვის.

პროგრამული სისტემის პრაქტიკული განხორციელება

პროგრამული სისტემა წარმოადგენს ვებ აპლიკაციას რომლის საშუალებითაც მომხმარებელს შეუძლია შემდეგი ქმედებების განხორციელება:

1. ტექსტის გაგზავნა
2. ფაილის გაგზავნა
3. რეგისტრაცია/ავტორიზაცია
4. ჯგუფური და ინდივიდუალური მიმოწერები
5. ფავორიტი ჩენელების არჩევა
6. User card
7. ემოჯი/გიფების გაგზავნა
8. ჯგუფების შექმნა(ღია, დახურული)
9. მიმოწერებში ფილტრაცია

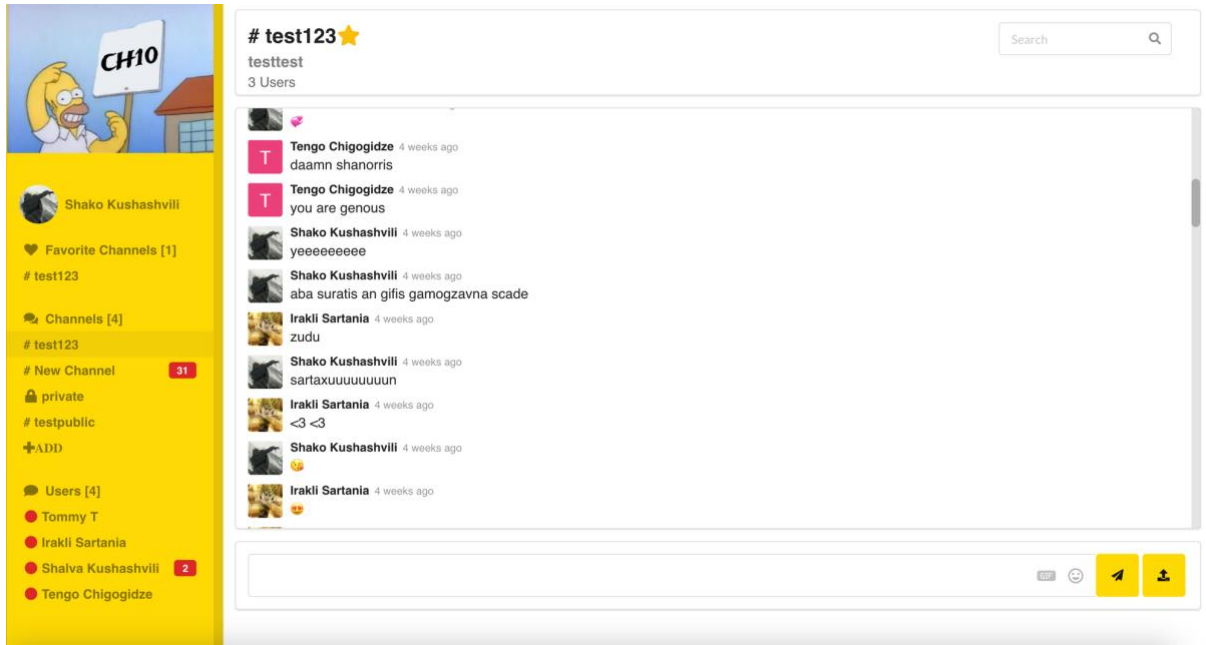
რეგისტრაცია/ავტორიზაცია



საიტზე შესვლისას მომხმარებელს ხვდება რეგისტრაცია/ავტორიზაციის გვერდი. ავტორიზაციის გავლისთვის საკმარისია მომხმარებლის მეილი და პაროლი. ამავედროულად, ავტორიზაციის გავლა შესაძლებელია Google Authentication-ის მეშვეობით,

რის საშუალებას Firebase გვაძლევს. Sign Up-ზე დაჭერის შემთხვევაში გამოჩნდება რეგისტრაციის ფორმა, სადაც მომხმარებელმა უნდა შეიყვანოს მისი სასურველი სახელი, მეილი და პაროლი. რეგისტრაციის გავლის შემდეგ აპლიკაცია მომხმარებელს ავტომატურად გადაიყვანს აპლიკაციის მთავარ კომპონენტში.

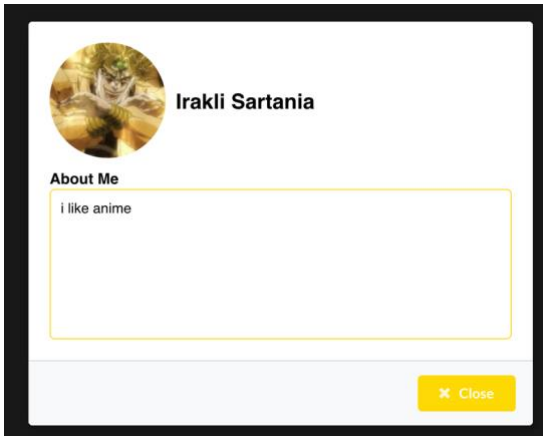
ჯგუფები და პირადი მიმოწერა



მომხმარებელი რეგისტრაცია/ავტორიზაციის შემდეგ გადამისამართდება მთავარ კომპონენტზე, სადაც არის ჩენელებისა და მომხმარებლების სია, და კონკრეტული ჩენელის კონტენტი. CH10-ში შესაძლებელია ჯგუფური, ასევე პირადი კონტაქტი.

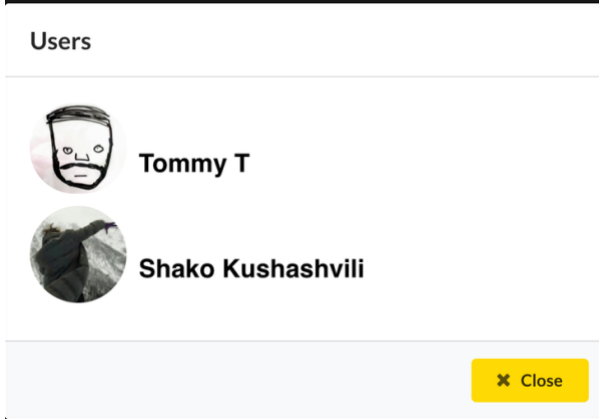
თითოეულ ჯგუფში აქვს ჯგუფის Description, ასევე ჯგუფში არსებული მომხმარებლების რაოდენობა. როდესაც მომხმარებელი დააჭერს მომხმარებლის რაოდენობას, ეკრანზე გამოვა ჯგუფში მყოფი მომხმარებლების სია. პირად მიმოწერაში მომხმარებლის სახელის ან სურათის დაჭერის შემდეგ გამოვა ამ მომხმარებლის ბარათი.

მომხმარებლებს აქვთ სტატუს ინდიკატორი (online ან offline), ასევე ყოველ ჩენელს და პირად მიმოწერას აქვთ notification-ების ფუნქციონალიც.



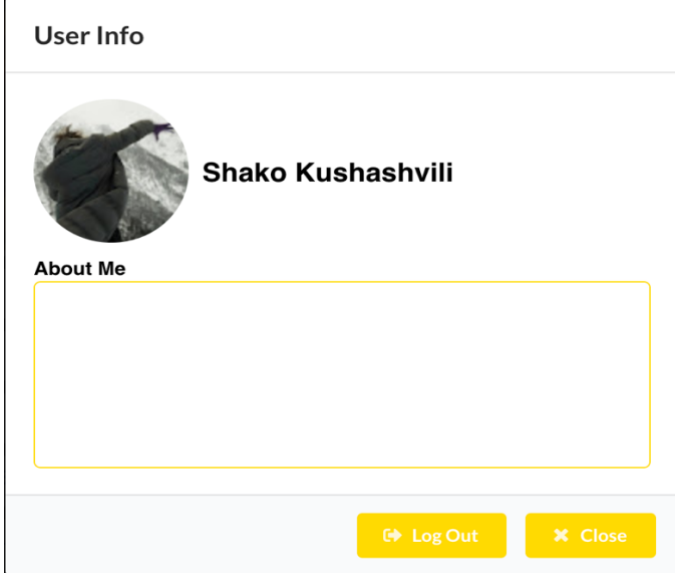
A user profile card for Irakli Sartania. It features a circular profile picture of a person with long hair, the name "Irakli Sartania", and an "About Me" section with the text "i like anime". A yellow "Close" button is located at the bottom right.

Users



A list of users. The first user is Tommy T, with a profile picture of a cartoon face. The second user is Shako Kushashvili, with a profile picture of a person in a dark jacket. A yellow "Close" button is at the bottom right.

მომხმარებლის ბარათი



A user info card for Shako Kushashvili. It features a circular profile picture of a person in a dark jacket, the name "Shako Kushashvili", and an "About Me" section. At the bottom, there are two yellow buttons: "Log Out" and "Close".

თუ მომხმარებელი დააკლიკებს თავის სურათს ან სახელს, ეკრანზე გამოვა მომხმარებლის ბარათი, სადაც მითითებულია მომხმარებლის შესახებ ინფორმაცია. ასევე, შესაძლებელია პროფილის სურათის შეცვლა სურათის დაკლიკების მეშვეობით.

ჯგუფის დამატება

Create Channel

Enter channel name

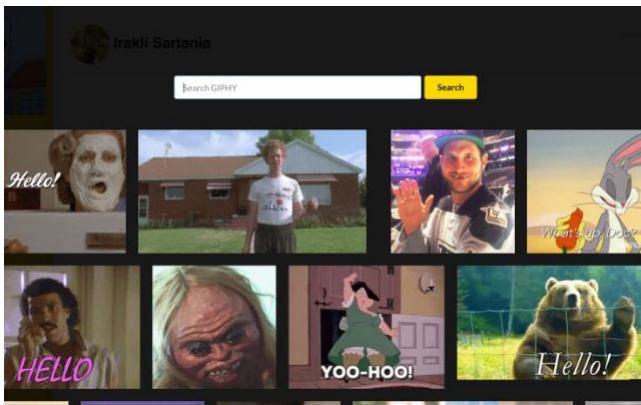
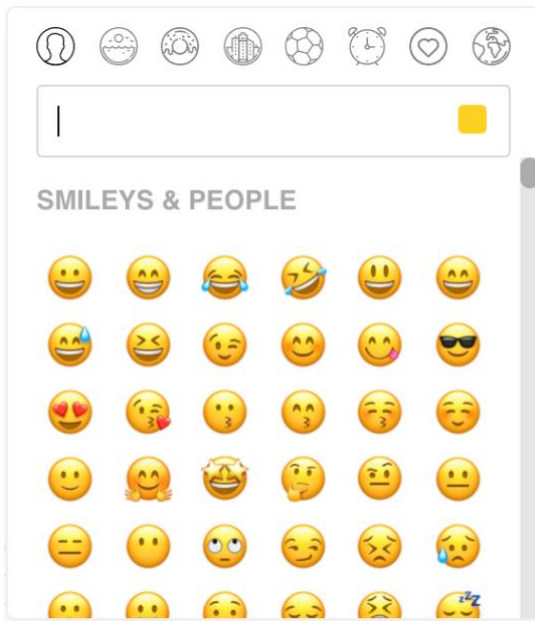
Enter channel description

Enter password (optional, only if you want a private channel)

+ Add × Cancel

მომხმარებელს შეუძლია ჯგუფის დამატება + ADD ღილაკის მეშვეობით, რომელიც მდებარეობს ჩენელების სიის ბოლოს. მომხმარებელმა უნდა შეიყვანოს ჩენელის სახელი, აღწერა. ასევე შესაძლებელია პირადი ჯგუფის შექმნა თუ მომხმარებელი ჩენელის შექმნისას ჩაწერს პაროლს.

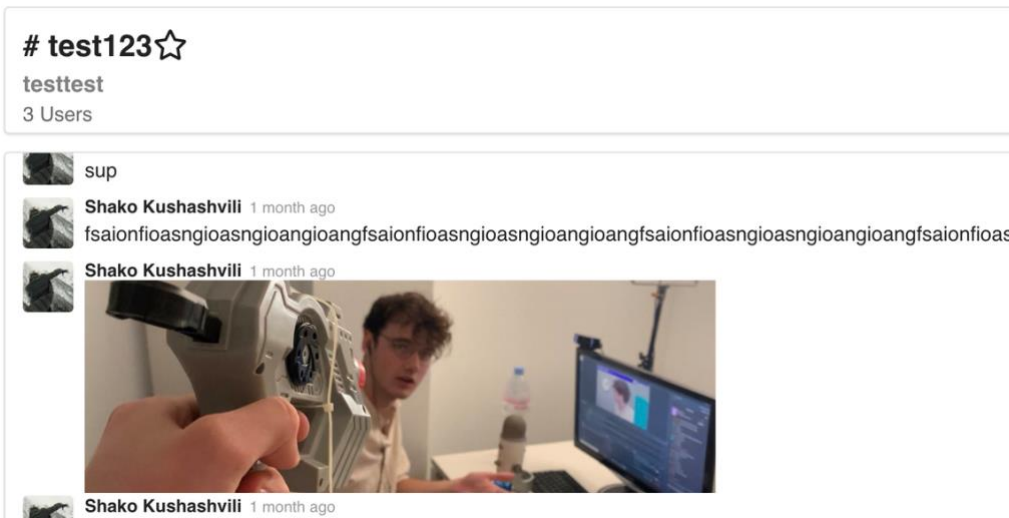
სმაილიკები/ემოჯი & გიფები



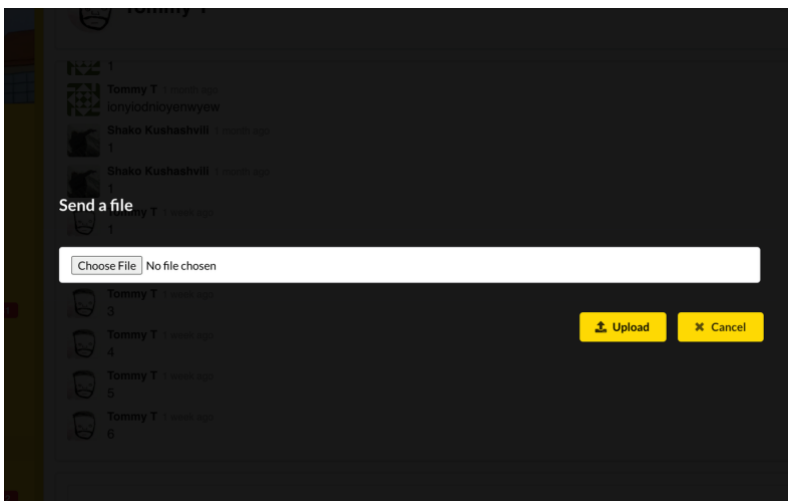
მომხმარებელს აქვს შესაძლებლობა არამხოლოდ ტექსტი, არამედ ემოჯი გამოიყენოს საკუთარი ემოციების გამოსახატად. ასევე, მომხმარებელს შეუძლია გიფების გაგზვნა. გიფების გვერდი გამოჩნდება GIF ლილაკის დაჭერის შემთხვევაში, ხოლო ემოჯის მინიფანჯარა გამოჩნდება სმაილიკის დაჭერის შემთხვევაში



ფაილების გაგზავნა

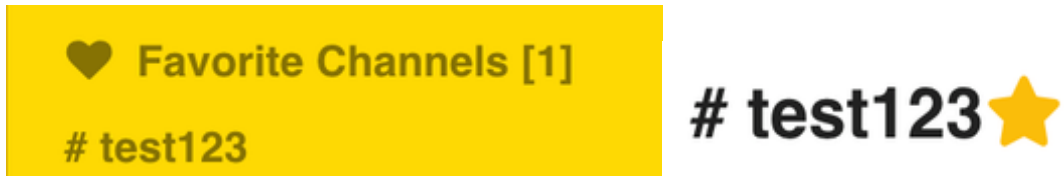


მომხმარებელს შეუძლია სურათის გაგზავნა Upload ღილაკის მეშვეობით, სურათის გაფართოება აუცილებლად უნდა იყოს .jpeg, .jpg, ან .png. სურათის დაჭერის შემთხვევაში მომხმარებელს შეუძლია სურათის ნახვა მთლიან ეკრანზე. მომხმარებლებს შეუძლია არამარტო სურათების, არამედ ვიდეოების(.mp4, .mov და .avi) და სხვა ფაილების გაგზავნაც.



საყვარელი ჯგუფის ან ჯგუფების აღნიშვნა

მომხმარებელს შეუძლია აირჩიოს საყვარელი ჯგუფი ვარსკვლავის დაჭერის მეშვეობით. საყვარელ ჩენელები გამოჩნდება Favorite Channels სიაში. ეს მომხმარებელს გაუმარტივებს მისი საყვარელი ჯგუფის კონტენტის ნახვას.



მიმოწერაში ძებნა

მომხმარებელს შეუძლია ნებისმიერი სახის ჯგუფში ან პირად მიმოწერაში შესარულოს მიმოწერების ძებნა Search input-ის მეშვეობით. მიმოწერები გაიფილტრება იმის მიხედვით, თუ რას ჩაწერს მომხმარებელი Search-ში.



დასკვნა

პროექტის მიზანი იყო შეგვექმნა ისეთი ჩათის ვებ აპლიკაცია, რომელიც იქნებოდა მომხმარებლისთვის მარტივად გამოყენებადი და ამავდროულად წერილების გაგზავნა მოხდებოდა რეალურ დროში სწრაფად და მოსახერხებლად. ამ მიზნით ჩვენს გარემოცვაში (მეგობრები, ნათესავები, სტუდენტები, პედაგოგები) ჩავატარეთ კვლევები, რომლიდანაც გავარკვიეთ მათი მოთხოვნები ჩათებთან და როგორ იყენებენ ისინი ჩათებს ყოველდღიურ ცხოვრებაში, რის შემდეგაც შევქმენით მომხმარებლის კმაყოფილებაზე დაფუძნებული UX.

UX-ის შექმნის შემდეგ აუცილებელი იყო იმის დადგენა თუ რა ფუნქციონალები შევიდოდა პროექტში. უპირველეს ყოვლისა ასეთი ტიპის ვებ აპლიკაციებს ესაჭიროებათ ავტორიზაცია/რეგისტრაცია. ავტორიზაციის გავლის შემდეგ მომხმარებელი შედის უკვე CH10-ის სივრცეში სადაც შესაძლებლობა აქვს სხვებთან დაიწყოს მიმოწერა როგორც პირადი ასევე ჯგუფური მიმოწერების სახით.

CH10ის მოდულების ძირითადი ნაწილი აწყობილია react ის საშუალებით. ავტორიზაციისთვის და მონაცემთა ბაზისთვის გამოვიყენეთ firebase, რადგან firebase გვამძლევს საშუალებას მომხმარებელს ავტორიზაცია გავატაროთ გუგლის აქაუნთის მეშვეობითაც. ბექის ნაწილში გამოვიყენეთ redux, რადგან redux კარგად მუშაობს reactთან. როუტინგისთვის გამოვიყენეთ react-routing. შედეგად მივიღეთ ისეთი ვებ აპლიკაცია, როგორც გვინდოდა რომ გამოგვესვლოდა.

არსებულ პროგრამულ სისტემაზე გვაქვს იმის შესაძლებლობა, რომ დავამატოთ ისეთი ფუნქციონალები, როგორცაა დარეკვა, მომხმარებლებისთვის შავი სიის დამატება, რათა სპამები აირიდონ თავიდან, მომხმარებელს მისცემს საშუალებას თავისი გემოვნებით მოაწყოს გარემო (შეცვალოს ფერი, შეცვალოს სურათი, ჩათის ბექგრაუნდის დაყენება), მესიჯებზე დარეაქტება, მომხმარებლის მოთაგვა და სხვა.

ლიტერატურა

1. <https://www.w3schools.com/> თანამედროვე ტერნოლოგიების გამოსაყენებლად
2. <https://stackoverflow.com/> პრობლემატური საკითხების გადასაჭრელად
3. <https://developers.giphy.com/> GIPHY API, რომლის მეშვეობით შევქმენით გიფების გაგზავნის ფუნქციონალი

დანართი

პროექტი ატვირთულია გითჰაბზე - <https://github.com/ShakNorris/CH10>