

**KLAIPĖDOS VALSTYBINĖ KOLEGIJA
TECHNOLOGIJŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA**

LEISTA GINTI

**INTERNETO SVETAINIŲ KŪRIMO KALBŲ PALYGINAMOJI
ANALIZĖ**

BAIGIAMASIS DARBAS

BD KVK 653I10002 I9-3

DIPLOMANTAS

VIDMANTAS VILKIUS

2017-06

VADOVAS

SIGITA GALDIKIENĖ

2017-06

RECENZENTAS

2017-06

KLAIPĖDA, 2017

SANTRAUKA

Vidmantas Vilkius (2017). Interneto svetainių kūrimo kalbų palyginamoji analizė. Informatikos profesinio bakalauro baigiamasis darbas. Studijų programos valstybinis kodas 653I10002. Vadovė: lektorė Sigita Galdikienė. Klaipėda: Klaipėdos valstybinė kolegija, Technologijų fakultetas, Informacinių technologijų katedra, 55 psl.

Baigiamojo darbo tikslas yra atlikti interneto svetainių kūrimo kalbų palyginamąją analizę. Baigiamajame darbe aptariama interneto svetainių kūrimo kalbų samprata, apžvelgiami interneto svetainių kūrimo principai, parengiama interneto svetainių kūrimo kalbų palyginamosios analizės atlikimo metodika, aprašoma interneto svetainių kūrimo kalbų palyginamosios analizės atlikimo eiga, apibendrinami ir aptariami rezultatai bei pateikiamos rekomendacijos dėl interneto svetainių kūrimo kalbų naudojimo.

PAGRINDINIAI ŽODŽIAI: internetas, interneto svetainė, kūrimo kalba.

SUMMARY

Vidmantas Vilkius (2017). Comparative Analysis of Website Development Languages. Professional Bachelor's thesis in Informatics. State code of the study programme: 653I10002. Academic supervisor: Lecturer Sigita Galdikienė. Klaipėda: Klaipėda State University of Applied Sciences, Faculty of Technologies, Department of Information Technologies, 55 pages.

The goal of the graduation thesis is to perform comparative analysis of website development languages. In the graduation thesis website development languages concepts are discussed, website development principles are reviewed, the methodology of website development language comparative analysis is prepared, website development languages comparative analysis is described, results are summarized and discussed, website development languages usage guidances are provided.

TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS

C

CSS (angl. *Cascading style sheet*) – kalba, skirta nusakyti kita struktūriniu kalba aprašyto dokumento vaizdavimą [20].

D

DNS (angl. *Domain Name System*) – internete be skaitinių **IP** adresų naudojami simboliniai vardai [22].

F

FPM (angl. *FastCGI Process Manager*) – **FastCGI** procesų valdiklis, naudojamas vykdyti **FastCGI PHP** svetainės kalboje [1].

H

HTML (angl. *Hypertext Markup Language*) – kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete [20].

HTTP (angl. *Hypertext Transfer Protocol*) – hipertekstu persiuntimo protokolas saityno duomenims (ištekliams) persiųsti [4].

M

MySQL – nemokama ir atviro kodo duomenų bazių valdymo sistema [9].

P

Programavimo kalba – dirbtinė kalba, skirta programoms rašyti [3].

S

SQL (angl. *Structured Query Language*) – sąrašinių duomenų bazių užklausų kalba [14].

T

Turinio tvarkymo sistema – kompiuterinė sistema dokumentams, esantiems tinkle, kolektyviai kurti ir tvarkyti [7].

U

Unicode – dviejų baitų kodas, kuris gali užkoduoti 65536 skirtingus simbolius [4].

X

XSS (angl. *Cross-site scripting*) – tai pažeidžiamumas, dažniausiai aptinkamas tinklalapiuose, kuris leidžia įterpti papildomą programinį kodą į naudotojų peržiūrimą puslapį [2].

Ž

Žiniatinklis – Trumpiausias ir dažniausiai vartojamas pasaulinio hipertekstinės informacijos tinklo (angl. *World Wide Web*) pavadinimas [18].

TURINYS

ĮVADAS	9
1. INTERNETO SVETAINIŲ KŪRIMO KALBOS SAMPRATA	10
1.1. Interneto samprata	10
1.2. Interneto svetainės samprata.....	10
1.3. Kūrimo kalbos samprata.....	11
1.4. Skyriaus apibendrinimas.....	12
2. INTERNETO SVETAINIŲ KŪRIMO PRINCIPAI.....	13
2.1. Interneto svetainės struktūra	13
2.2. Interneto svetainės kūrimo būdai.....	14
2.2.1 Svetainės kūrimas <i>HTML</i> pagrindu	14
2.2.2 Svetainės kūrimas naudojantis turinio tvarkymo sistema.....	16
2.3. Skyriaus apibendrinimas.....	17
3. INTERNETO SVETAINIŲ KŪRIMO KALBŲ PALYGINAMOSIOS ANALIZĖS ATLIKIMO METODIKA.....	18
3.1. Interneto svetainių kūrimo kalbų apžvalga ir palyginimas.....	18
3.1.1 Interneto svetainių kūrimo kalba <i>PHP</i>	18
3.1.2 Interneto svetainių kūrimo kalba <i>Ruby</i>	21
3.1.3 Interneto svetainių kūrimo kalba <i>Python</i>	23
3.1.4 Interneto svetainių kūrimo kalbų apžvalgos apibendrinimas	25
3.2. Interneto svetainių kūrimo kalbų analizės atlikimo kriterijai ir vertinimo skalės	26
3.3. Interneto svetainių kūrimo kalbų palyginamosios analizės bandymų atlikimo aplinka	28
3.4. Interneto svetainių kūrimo kalbų palyginamosios analizės bandymų procesai.....	29
3.5. Skyriaus apibendrinimas.....	31
4. INTERNETO SVETAINIŲ KŪRIMO KALBŲ PALYGINAMOSIOS ANALIZĖS ATLIKIMO EIGA, REZULTATŲ APIBENDRINIMAS IR APTARIMAS.....	32
4.1. Interneto svetainių kūrimo kalbų palyginamosios analizės atlikimo eiga.....	32
4.1.1 Interneto svetainės kūrimo kalbos <i>PHP</i> bandymai.....	33
4.1.2 Interneto svetainės kūrimo kalbos <i>Python</i> bandymai	37
4.1.3 Interneto svetainės kūrimo kalbos <i>Ruby</i> bandymai	39
4.2. Interneto svetainių kūrimo kalbų palyginamosios analizės bandymų rezultatai	42
4.3. Skyriaus apibendrinimas.....	50
5. REKOMENDACIJOS DĖL INTERNETO SVETAINIŲ KŪRIMO KALBŲ NAUDOJIMO IŠVADOS IR PASIŪLYMAI	52
IŠVADOS	53
INFORMACIJOS ŠALTINIŲ SĄRAŠAS.....	54

PRIEDAI	56
---------------	----

LENTELIŲ SĄRAŠAS

1 lentelė. Interneto svetainių struktūrų tipai	13
2 lentelė. Svetainės kūrimo palyginimo kriterijai.....	14
3 lentelė. Svetainės kūrimo kalbos ir paskirtis [7].....	15
4 lentelė. <i>HTML</i> pagrindu svetainės palyginimo rezultatas	15
5 lentelė. Turinio valdymo sistemos palyginimo rezultatas	16
6 lentelė. Svetainės kūrimo būdų palyginimo rezultatai.....	17
7 lentelė. Interneto svetainių kūrimo apžvalga ir palyginimas	18
8 lentelė. Interneto svetainių kalbų palyginimas	25
9 lentelė. Interneto svetainių kūrimo kalbų kriterijai ir įvertinimai	26
10 lentelė. Bandymo priemonių apžvalga	28
11 lentelė. Interneto svetainių kūrimo kalbų bandymų atlikimo matrica	30
12 lentelė. <i>PHP</i> svetainės užklausų vidurkis vykdant 100 užklausų.....	34
13 lentelė. Duomenų bazės poveikis <i>PHP</i> svetainės užklausų vykdymui	35
14 lentelė. Procentinis duomenų bazės poveikis <i>PHP</i> svetainės užklausų vykdymui	35
15 lentelė. <i>PHP</i> svetainės apdorojamų užklausų poveikis procesoriui vykdant 100 užklausų	35
16 lentelė. <i>Python</i> svetainės užklausų vidurkis vykdant 100 užklausų	37
17 lentelė. Duomenų bazės poveikis <i>Python</i> svetainės užklausų vykdymui.....	38
18 lentelė. Procentinis duomenų bazės poveikis <i>Python</i> svetainės užklausų vykdymui.....	38
19 lentelė. <i>Python</i> svetainės procesoriaus galios naudojimas vykdant 100 užklausų	38
20 lentelė. <i>Ruby</i> svetainės užklausų vidurkis vykdant 100 užklausų	40
21 lentelė. Duomenų bazės poveikis <i>Ruby</i> svetainės užklausų vykdymui.....	40
22 lentelė. Procentinis duomenų bazės poveikis <i>Ruby</i> svetainės užklausų vykdymui.....	41
23 lentelė. <i>Ruby</i> svetainės procesoriaus galios naudojimas vykdant 100 užklausų	41
24 lentelė. Interneto svetainių kūrimo kalbų bandymų rezultatai	42

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav. Interneto svetainių kūrimo kalbos <i>PHP</i> logotipas [9]	19
2 pav. Interneto svetainių kūrimo kalbos <i>Ruby</i> logotipas [12]	21
3 pav. Interneto svetainių kūrimo kalbos <i>Python</i> logotipas [16]	23
4 pav. Interneto svetainės struktūra	32
5 pav. Duomenų bazės struktūra	33
6 pav. <i>PHP</i> programavimo kalbos svetainės kodo tipai ir eilučių skaičius	36
7 pav. <i>Python</i> programavimo kalbos svetainės kodo tipai ir eilučių skaičius	39
8 pav. <i>Ruby</i> programavimo kalbos svetainės kodo tipai ir eilučių skaičius	42
9 pav. Užklausų apdorojimo kiekis per 1 sekundę kompiuteryje <i>Samsung NP350V5C</i>	44
10 pav. Užklausų apdorojimo kiekis per 1 sekundę kompiuteryje <i>Lenovo g510</i>	44
11 pav. Užklausų apdorojimo kiekis per 1 sekundę kompiuteryje <i>Hp 15-n245sa</i>	45
12 pav. Apdorojamų užklausų kiekio rezultatai	45
13 pav. Duomenų bazės poveikis užklausų vykdymui kompiuteryje <i>Samsung NP350V5C</i>	46
14 pav. Duomenų bazės poveikis užklausų vykdymui kompiuteryje <i>Lenovo g510</i>	47
15 pav. Duomenų bazės poveikis užklausų vykdymui kompiuteryje <i>Hp 15-n245sa</i>	47
16 pav. Apdorojamų užklausų poveikis kompiuterio <i>Samsung NP350V5C</i> procesoriui	48
17 pav. Apdorojamų užklausų poveikis kompiuterio <i>Lenovo g510</i> procesoriui	49
18 pav. Apdorojamų užklausų poveikis kompiuterio <i>Hp 15-n245sa</i> procesoriui	49
19 pav. Apdorojamų užklausų poveikis procesoriams rezultatai	50

IVADAS

Yra daugybė straipsnių kuriuose aiškinama svetainių kūrimo būdų bei kaip jas kurti, tačiau svetainės kūrimo principai nėra tiksliai apibrėžti, susisteminti, nepaaiškinamos kiekvieno etapo žingsniai. Šiame darbe siekiama išsiaiškinti kokie yra svetainių kūrimo principai ir kaip jie įtakoja galutinį darbo rezultatą.

Darbo objektas: interneto svetainių kūrimo kalbos.

Baigiamojo darbo tikslas: atlikti interneto svetainių kūrimo kalbų palyginamąją analizę.

Siekiant įgyvendinti numatytą tikslą, keliami šie baigiamojo darbo uždaviniai:

1. Aptarti interneto svetainių kūrimo kalbos sampratą;
2. Apžvelgti interneto svetainių kūrimo principus;
3. Parengti interneto svetainių kūrimo kalbų palyginamosios analizės atlikimo metodiką;
4. Aprašyti interneto svetainių kūrimo kalbų palyginamosios analizės atlikimo eigą, apibendrinti ir aptarti rezultatus;
5. Pateikti rekomendacijas dėl interneto svetainių kūrimo kalbų naudojimo.

Programavimo kalbos: *PHP, Python, Ruby*.

Baigiamajam darbui parengti naudota programinė įranga:

1. *Notepad++* – programų kūrimo terpė (nemokama);
2. *WampServer 3.0.6* – tinklapių kūrimo terpė (nemokama);
3. *Rack* – tinklapių kūrimo terpė (nemokama);

Metodika: literatūros analizė, internetinių šaltinių informacijos sisteminimas, kriterijų planavimas, duomenų lyginimas ir papildomos rekomendacijos.

1. INTERNETO SVETAINIŲ KŪRIMO KALBOS SAMPRATA

Šiame skyriuje, remiantis surinkta ir susisteminta moksline literatūra, aptariamos reikšmės: **interneto**, **internetinės svetainės** ir **kūrimo kalbos** sampratas. Sampratų aptarimai ir apibendrinimai pateikiami poskyriuose:

1. interneto samprata;
2. interneto svetainės samprata;
3. kūrimo kalbos samprata;
4. skyriaus apibendrinimas.

1.1. Interneto samprata

Pagal aiškinamąjį elektrotechnikos terminų žodyną [22] internetas yra pasaulinė kompiuterių tinklų sistema, jungianti visuotinius ir vietinius kompiuterių tinklus. Šie tinklai yra jungiami įvairiomis ryšio linijomis ir maršrutizatoriais, kurie iš vieno tinklo gautus duomenų paketus modifikuoja ir nukreipia į kitą tinklą.

Naudojantis *Enciklopedinu kompiuterijos žodynu* galime teikti kad internetas yra pasaulinis kompiuterių tinklas, jungiantis kitus tinklus ir veikiantis **TCP/IP** protokolo pagrindu [4]. Nutraukus darbą bet kuriam kompiuteriui arba grupei kompiuterių, internetas išlieka gyvybingas. Šis tinklas yra valdomas ir tvarkomas decentralizuotai – neturi vieno šeimininko. Internetas neturi centralizuoto atstovybės, kuri reguliuotų technologijų panaudojimą ar taisyklių dėl jo prieigos ir naudojimo – kiekvienas tinklas nustato savo politiką. Interneto tiekėjai **ICANN** reguliuoja tik interneto protokolų adresų ir **DNS** (angl. *Domain Name System*) pavadinimus.

Duomenys tiek pačiame kompiuteryje, tiek siunčiant juos iš vieno kompiuterio į kitą kompiuterių tinkle, paverčiami **Unicode** dvejetainiu kodu. Informacija perduodama elektriniais signalais, kiekvienam bitui skiriamas tam tikras laiko tarpas, per kurį signalas perduodamas arba neperduodamas. Tokiu būdu perduodama ne tik tekstinė informacija, bet ir grafinė, muzikinė ir kiti tipai.

1.2. Interneto svetainės samprata

Pagal internetinį straipsnį *Interneto svetainė* [19], interneto svetainė apibūdinama kaip interneto tinklalapių rinkinys, kuriame kaupiama vaizdinė, garsinė ir tekstinė informacija. Pagal šį šaltinį interneto svetainė yra „šiuolaikinė vizitinė kortelė, kurioje įmonės ar žmonės pristato savo

darbus, pasiekimus, žmonių atsiliepimus bei pateikia savo kontaktus“. Ši priemonė svarbi kiekvienam, kuris nori tinkamai save reprezentuoti prieš kitus.

Interneto svetainė yra kolekcija internetinių tinklalapių, dažniausiai identifikuojamu su bendru domeno vardu, paskelbtų ant vieno internetinio serverio. Interneto svetainė gali būti pasiekama per internetinio protokolo serverį kaip internetas ar privatų vietinį tinklą. Visos viešai pasiekiamos interneto svetainės kolektyviškai sudaro **žiniatinklį** (angl. *World Wide Web*).

interneto svetainėje **W3C** pagrindiniai svetainės statybiniai blokai yra įvardijami kaip interneto tinklalapiai [20]. Interneto tinklalapiai yra dokumentai, parašyti su **HTML** (angl. *Hypertext Markup Language*) programavimo kalba ir išverčiami naudojama naršykle. Jie yra skirstomi į statinius ir dinامينius. Statiniai tinklalapiai vaizduoja tą patį turinį kiekvieną kartą naudojant informaciją, o dinaminų puslapių turinys gali kisti kiekvieną kartą apsilankius. Tinklalapyje esantis scenarijus vykdo serverio funkcijas, kurios grąžina informaciją kaip datą, laiką ir duomenų bazės informaciją. Visi duomenys yra gražinami **HTML** kalba, tai kai tinklalapis yra pajungiamas internetine naršykle, naršyklei tereikia išversti numatytą kalbą.

1.3. Kūrimo kalbos samprata

Prieš tai aptarėme kas yra interneto svetainė, dabar aptarsime jos pagrindą – kūrimo kalbą. Kūrimo kalba yra kalba, kuria yra kuriamas tam tikros srities projektas. Šiuo atveju kūrimo kalba svetainėse yra **programavimo kalba**, kuri kontroliuoja programos veiksmus. Ši kalba laikoma tinkamai sukonstruota kai su sukurtomis komandomis galima bendrauti su kompiuterio sistema.

Programavimo kalba paprastai yra padalyta į du komponentus: semantiką ir sintaksę. Sintaksės komponentas sukuria kodo formą ar tipą, o semantika suteikia tam prasmę. Kiekviena programavimo kalba yra skirtinga, jos gali turėti skirtingas paskirtis; gali būti naudojamos kuriant programinę įrangą, kitos – bendrauti su duomenų bazėmis ar kurti svetaines. Taigi programavimo kalba galima laikyti kaip anotacija, kuri padeda rašyti programas, kitaip dar žinomus kaip algoritmus.

Ne visos programavimo kalbos gali būti naudojamos kaip svetainės kūrimo kalbos. Naudojantis **Cleverism** internetine svetaine programavimo kalba kuriamos svetainės turi atitikti tam tikrus faktorius [3]:

- orientuota platforma – platforma, kurioje veiks sukurta programa. Programavimo kalba negali veikti ant visų esamų platformų todėl renkantis kalbą reikia įsitikinti ant kokių platformų ji gali veikti;

- domeno kalbos atitikimas – kalba turi būti pasirenkama atsižvelgiant į jūsų turimas problemines sritis. Domeno kalba yra sukurta išspręsti problemas tik tam tikrame domene, kuri gali neveikti kituose domenuose;
- efektyvumas – kompiliatorius, kuris dirba su pasirinkta programavimo kalba, turi būti našus kad programavimo kalba galėtų atlikti darbą greitai;
- elastingumas – programavimo kalba turi būti pakankamai lanksti, kad ją būtų galima papildyti naujomis programomis ir funkcijomis;
- bibliotekų prieinamumas – pasirinkta kalba turi turėti biblioteką, kuri yra pajėgi išspręsti iškilusias problemas su programavimo kalba;
- projekto dydis – projektai gali būti dideli ir maži. Svarbu pasirinkti kalbą, kuri gali palaikyti projekto dydį;
- gamybos laikas – laikas, per kurį yra sukurama programa. Su pasirinkta kalba kodavimas turi būti nesudėtingas ir greitas;
- įrankiai – programavimo kalba gali turėti įvairių įrankių, kurie gali palengvinti kodavimą, redagavimą ir valdymą.

Jei pasirinkta programavimo kalba neatitinka vieną ar daugiau prieš tai išvardytų kriterijų, ji gali būti netinkama svetainių kūrimui, gali neatlikti tam tikrų funkcijų arba apskritai nefunkcionuoti.

1.4. Skyriaus apibendrinimas

Aptarus interneto, interneto svetainės ir kūrimo kalbos sampratą, nustatyta kad internetas yra pasaulinė kompiuterių tinklų sistema, jungianti visuotinius ir vietinius kompiuterių tinklus. Šio tinklo duomenys yra paverčiami **Unicode** dvejetainiu kodu. Nustatyta, kad interneto svetainė yra interneto tinklalapių rinkinys, kuriame kaupiama vaizdinė, garsinė ir tekstinė informacija. Interneto svetainės yra parašytos naudojantis **HTML** programavimo kalba, interneto tinklalapiai yra skirstomi į dvi grupes – statinius ir dinامينius. Kūrimo kalba yra kalba, kuria yra kuriamas tam tikros srities projektas, dažnai asocijuojama su programavimo kalba, naudojama kurti svetaines. Programavimo kalba yra padalyta į semantinę ir sintaksinę komponentus. Kuriant Interneto svetaines programavimo kalbos turi atitikti kriterijus, leidžiančius atlikti nurodytus veiksmus.

Aptarus atskiras sampratą suprasta, kad **internetu svetainių kūrimo kalba** yra programavimo kalba, naudojama kuriant tinklalapių rinkinį skirtą pasaulinėje kompiuterių tinklų sistemoje internete.

2. INTERNETO SVETAINIŲ KŪRIMO PRINCIPAI

Antrajame skyriuje aptariamos svetainių kūrimo principai. Pradedant kurti svetainę yra labai svarbu ją kruopščiai suplanuoti. Kuriant internetinę svetainę jos sudėtinės dalys yra:

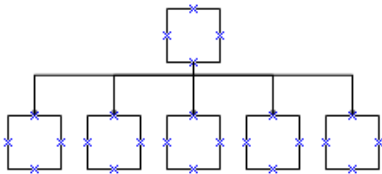
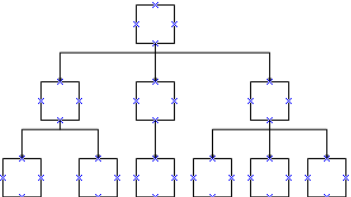
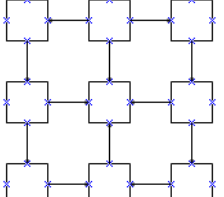
- interneto svetainės struktūra;
- interneto svetainės kūrimo būdai.

2.1. Interneto svetainės struktūra

Svetainių kūrimas pažymi kad pagrindinis kiekvienos interneto svetainės elementas yra jos pradinis puslapis. Pirmajame svetainės tinklalapyje pateikiama informaciją apie interneto svetainės turinį, jos struktūrą. Turinio aprašymą sudaro atskirų svetainės skyrių tinklalapius atverčiantys saitai. Taip sukurama patraukli ir patogi analizei bei duomenų paieškai hierarchinė interneto svetainės struktūra.

Svetainės struktūra turi būti paprasta ir suprantama, negali turėti daug sluoksnių kad naudotojai nesugaištų laiko ieškant reikalingos informacijos. Tinklalapiui nuolatos augant jo struktūra tampa sudėtingesnė. Kuriant tinklalapį gali būti naudojamos įvairios svetainės struktūros. Pagal Patrick J. Lynch ir Sarah Horton interneto svetainę *Site structure* [11], struktūros yra skirstomos į tris tipus: linijinį, hierarchinį ir tinklinį. Interneto svetainių Struktūrų tipai pavaizduoti 1 lentelėje.

1 lentelė. Interneto svetainių struktūrų tipai

Tipas	Linijinė	Hierarchinė	Tinklinė
Struktūra			

Šioje lentelėje pavaizduotos trys pagrindinės svetainės struktūros: linijinė, hierarchinė ir tinklinė. Kiekviena iš šių struktūrų pasižymi savo privalumais:

- linijinėje struktūroje nuorodos juda tik pirmyn ir atgal. Linijinėje struktūroje lengva susigaudyti, nes yra nedaug naršymo galimybių, tačiau dėl to nukenčia duomenų pateikimas, naudotojas ribojamas šio tipo sistemoje;

- naudotojui patraukliausias būdas susisteminti interneto dokumentus yra naudoti hierarchinę struktūrą. Hierarchinis tipas naudoja sudėtingesnę struktūros tipą. Dauguma interneto pagalbos sistemų yra hierarchinio tipo;
- tinklinė svetainė naudoja struktūrą, kuri lankytoji suteikiama galimybė nevaržomai keliauti per puslapius iš bet kurios interneto svetainės pozicijos, tačiau tokia struktūra neretai sukelia sąmyšį;

Nors visos trys struktūros pasižymi savais pranašumais, didesnės svetainės naudoja visas tris struktūras. Svetainės pagrindinė struktūra yra hierarchinė, tačiau dėl daugybės šalutinių nuorodų svetainė įgauna tinklinę išvaizdą. Linijinė struktūra naudojama tik tose vietose, kur nenorima, kad naudotojai praleistų svarbią informaciją.

2.2. Interneto svetainės kūrimo būdai

Interneto svetainių kūrimo būdai yra įvairūs, bet dažniausiai naudojami du – svetainės kūrimas nuo pagrindų naudojantis svetainių kūrimo kalbas, arba naudojantis **turinio tvarkymo sistemas** (angl. *Content Management System*). Šie būdai yra skirtingi, todėl planuojant interneto svetainę svarbu pasirinkti kuris interneto svetainės kūrimo būdas bus naudojamas. Norint išsiaiškinti tinkamesnį svetainės kūrimo būdą, reikalinga išsiaiškinti palyginimo kriterijus kuriais jie bus vertinami. 2 lentelėje pavaizduoti palyginimo kriterijai.

2 lentelė. Svetainės kūrimo palyginimo kriterijai

Kriterijai	Kriterijų aprašymas
Programavimo patirtis	Reikiamos programavimo kalbų žinios norint sukurti interneto svetainę
Lankstumas	Svetainės kūrimo limitas
Kūrimo būdas	Būdas, kuriuo sukurta interneto svetainė
Įdiegimas	Reikalingos įdiegti programinės įrangos
Įskiepai	Galimybė naudotis įskiepais interneto svetainėje
Atnaujinimas	Svetainės atnaujinimas reaguojant į išorinius pokyčius

Naudojantis šioje lentelėje išvardytus kriterijus bus palyginami skirtingi svetainių kūrimo būdai.

2.2.1 Svetainės kūrimas *HTML* pagrindu

Svetainės kūrimo kalba yra programavimo kalba, skirta svetainių kūrimui. Pagrindinėmis svetainių kūrimo kalbomis yra laikomos *HTML* ir *CSS*. *HTML* yra kalba, kuri apibūdina puslapio struktūrą naršyklei. *HTML* nurodo ką atvaizduoti puslapyje, bet nepaaiškina kaip. Tai atlieka *CSS*

programavimo kalba. Šios kalbos leidžia sukurti paprastas interneto svetainės, tačiau norint sukurti sudėtingesnes reikalingos papildomos svetainių kūrimo kalbos.

Renkantis svetainės kūrimo kalbą svarbu prisiminti, kad geriausia kūrimo kalba neegzistuoja. Kiekvienai svetainei yra pasirenkama kalba, kuri labiausiai tinka svetainės paskirčiai. Svetainių kūrimo kalbos dažniausiai skirstomos į tris dalis pagal jų paskirtį: kliento dalies, serverio dalies ir duomenų bazės dalį. Dalis daugiau žinomų skirtingų sričių svetainės kūrimo kalbų yra išvardytos **Upwork** svetainėje [8]. Dalis populiariesnių kalbų yra pavaizduotos 3 lentelėje.

3 lentelė. Svetainės kūrimo kalbos ir paskirtis [8]

Sritis	Kalba	Aprašymas
Kliento skripto kalbos	JavaScript	JavaScript yra programavimo kalba, su kuria kuriama animacija, žaidimai, programos ir panašaus tipo efektus.
	ActionScript	ActionScript yra kalba, naudojama Adobe Flash multimedijoje, kuri tinka interneto aplikacijoms, kurios naudoja Flash animaciją, garso ir video transliacijas.
Serverio skripto kalbos	PHP	PHP yra atviro kodo kalba palaikoma įvairių tipo platformose. Dažniausiai naudojama interneto svetainėse su mažu eismu.
	Java	Java yra serverio scenarijaus kalba naudojama didelio masto svetainėse su dideliu eismu.
	Python	Python yra bendrosios paskirties, aukšto lygio programavimo kalba, kuri daug dėmesio skiriama kodas aiškumui. Tai leidžia padaryti daugiau su mažiau kodo eilučių nei palyginus su kitomis kalbomis.
	Ruby	Ruby yra panaši į Python tuo, kad ji buvo sukurta tam, kad programavimas būtų našesnis sukuriant trumpą, paprastą, glaustą, nuoseklų bei lankstų kodą.
Duomenų bazės kalbos	SQL	SQL yra struktūrizuota užklausų kalba skirta bendrauti su duomenų baze.
	XQuery	Xquery skirta valdyti XML duomenis – ne tik XML failus, bet kokio tipo XMI duomenis, įskaitant duomenų bazes.

Tai tik maža dalis naudojamų kalbų, tačiau šios yra dažniausiai pasirenkamos kuriant interneto svetaines. Iš lentelės matyti, kad pagal pasirinktą svetainės kūrimo sritį gali būti naudojamos skirtingos kalbos.

Norint teisingai palyginti **HTML** pagrindu svetainės kūrimo būdą, naudojami 2 lentelės kriterijai. HTML pagrindu interneto svetainės palyginimo rezultatas yra pavaizduotas 4 lentelėje.

4 lentelė. **HTML** pagrindu svetainės palyginimo rezultatas

Kriterijai	Kriterijų aprašymas
Programavimo patirtis	HTML ir CSS bei kitos kalbos
Lankstumas	Neribotas
Kūrimo būdas	Programavimas
Įdiegimas	Reikalingas naudojant papildomas kalbas
Įskiepai	Nėra
Atnaujinimas	Savarankiškas svetainės atnaujinimas

Aukščiau esančioje lentelėje pavaizduota **HTML** pagrindu kuriamo interneto svetainės vertinimo lentelė. Pagal ją matyti, kad šiuo būdu kuriant interneto svetainę reikalingos

programavimo kalbų žinojimas. Svetainė yra programuojama, ji neturi įskiepių ar automatinių atnaujinimų. Tačiau svetainė gali būti neribotai modifikuojama pagal kūrėjo poreikius.

2.2.2 Svetainės kūrimas naudojantis turinio tvarkymo sistema

Paprastesnis svetainių kūrimo būdas yra turinio tvarkymo sistema. Turinio tvarkymo sistema (**TTS**) – įvairūs programiniai įrankiai, supaprastinantys informacinių sistemų turinio valdymą taip, kad sukuriant bei keičiant turinį ar jo struktūrą nereikalingos programavimo žinios. Paprastesnį apibūdinimą parašė Giedrius Gliožeris straipsnyje ***Turinio valdymo sistemos (TVS)*** [7] – tai „sąsaja tarp jūsų ir tinklalapio, padedanti kurti, saugoti ir redaguoti informaciją“. Vienas pagrindinių turinio valdymo sistemos tikslų – užtikrinti sklandų naudotojų darbą. Taikoma interneto svetainėms, laikraščių leidyboje, atviro turinio projektuose, įmonių valdyme, elektroninėje prekyboje ir panašiose srityse.

TTS sudaro dvi programas: turinio tvarkymo programa ir turinio pristatymo programa. Turinio tvarkymo programa dirba su turinio duomenų baze. Jos programiniai įrankiai padeda kurti bei formatuoti turinio elementus bei nustatyti jų struktūras ir sąsajas. Turinio pristatymo programa paima turinį iš duomenų bazės ir pateikia jį publikavimui apipavidalinus reikiamu formatu. Visa tai vyksta automatiškai, šios dalies sistemos naudotojas nesupranta ir nelaiko atskira programa.

Siekiant palyginti turinio valdymo sistemos svetainės kūrimo būdą, naudojami 2 lentelės kriterijai. Turinio valdymo sistemos palyginimo rezultatas yra pavaizduotas 5 lentelėje.

5 lentelė. Turinio valdymo sistemos palyginimo rezultatas

Kriterijai	Kriterijų aprašymas
Programavimo patirtis	Nereikalinga
Lankstumas	Ribotas
Kūrimo būdas	Vizuali sąsaja
Įdiegimas	Turinio valdymo sistema
Įskiepijai	Yra
Atnaujinimas	Automatinis

Kuriant svetainę naudojantis turinio tvarkymo sistema, programavimo žinios nėra reikalingos, svetainė kuriama per vizualų konstruktorį, kuriame kūrėjas gali naudotis specialiais įskiepiais ir įrankiais, svetainė automatiškai atnaujinama. Vis dėl to šis būdas turi trūkumų: svetainės modifikavimas yra ribotas, galima naudotis tik jau sukurtus dizainus, reikalinga instaliuoti programinę įrangą.

2.3. Skyriaus apibendrinimas

Aptarus interneto svetainių kūrimo principus buvo sužinota, kad interneto svetainės struktūra skirstoma į tris grupes: linijinę, hierarchinę ir tinklinę. Pagrindiniai interneto svetainės kūrimo būdai yra svetainės kūrimas **HTML** pagrindu ir naudojantis turinio tvarkymo sistema. Šie du skirtingi svetainių kūrimo būdai palyginti naudojantis kriterijais: reikalingos kalbos, lankstumu, kūrimo būdu, įdiegimu, įskiepimais prieinamumu ir atnaujinimo būdu. Svetainės kūrimo palygimo rezultatai yra pateikiami 6 lentelėje.

6 lentelė. Svetainės kūrimo būdų palyginimo rezultatai

Kriterijai	Svetainės kūrimas HTML pagrindu	Svetainės kūrimas turinio tvarkymo sistema
Programavimo Patirtis	HTML ir CSS bei kitos kalbos	Nereikalinga
Lankstumas	Neribotas	Ribotas
Kūrimo būdas	Programavimas	Vizuali sąsaja
Įdiegimas	Reikalingas naudojant papildomas kalbas	Turinio valdymo sistema
Įskiepijai	Nėra	Yra
Atnaujinimas	Savarankiškas svetainės atnaujinimas	Automatinis

Iš lentelės matyti kad svetainės kūrimas **HTML** pagrindu leidžia didesnę modifikavimo laisvę nei turinio tvarkymo sistema tačiau antrasis variantas nereikalauja papildomų žinių ir turi pagalbinių objektų svetainės kūrimui pagreitinanti.

3. INTERNETO SVETAINIŲ KŪRIMO KALBŲ PALYGINAMOSIOS ANALIZĖS ATLIKIMO METODIKA

Trečiame skyriuje rengiama interneto svetainių kūrimo kalbų palyginamosios analizės atlikimo metodika. Ši metodika yra išskaidoma į šias interneto svetainių kūrimo kalbų grupes:

1. apžvalga ir palyginimas;
2. analizės atlikimo kriterijai ir vertinimo skalės;
3. analizės bandymų atlikimo aplinka;
4. palyginamosios analizės bandymų procesai

Šiame darbe tiriamos ir analizuojamos **PHP**, **Python** ir **Ruby** svetainių kūrimo kalbos, jų pranašumai, trūkumai ir savybės. Šiai analizei atlikti yra reikalingos priemonės kaip kompiuteriai, šiuose įrenginiuose įdiegtos programinės įrangos. Norint kad darbas būtų išsamus, yra detalizuojamas atliekamų bandymų planas, pateikiama būsima palyginamosios analizės atlikimo eiga, bei išdėstomi procesai su interneto svetainių kalbų palyginimu.

3.1. Interneto svetainių kūrimo kalbų apžvalga ir palyginimas

Interneto svetainių kūrimo kalbos yra programavimo kalbos, kurios yra tinkamos interneto svetainių kūrimui. Interneto svetainių kūrimo kalbų analizei buvo pasirinktos trys serverio skripto kalbos: **PHP**, **Ruby** ir **Python**. Norint palyginti šias tris kalbas yra pasirinkti kriterijai, pavaizduoti 7 lentelėje.

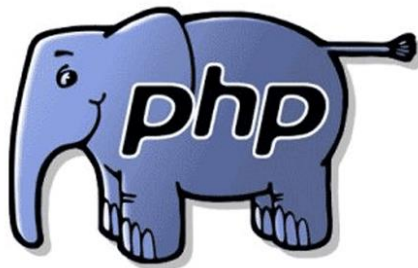
7 lentelė. Interneto svetainių kūrimo apžvalga ir palyginimas

Nr.	Kriterijai	Apibūdinimas
1.	Funkcijų kiekis	Funkcijos ir jų kiekis pateikiamas oficialiuose programavimo kalbų dokumentacijose.
2.	Kodo tipas	Interneto svetainės kalba yra atviro kodo ar uždaro.
3.	Saugumas	Programavimo kalbos saugumas prieš iškylančias grėsmes.
4.	Palaikomos platformos	Platformos, kurias palaiko numatytoji kalba.

Šios kalbos bus lyginamos naudojantis 7 lentelėje iškeltais kriterijais: kalbos sudėtingumas, mokymasis, sukurto kodo saugumas, kodo tipas ir palaikomos platformos.

3.1.1 Interneto svetainių kūrimo kalba **PHP**

PHP yra dinaminė interpretuojama programavimo kalba. Šią kalbą sukūrė Rasmus Lerdorf'as 1995 metais interneto svetainių kūrimui. **PHP** (žr. 1 pav.) kalba yra skriptų kalba naudojama serveriuose, kuriose yra **Apache** serverio įranga. Sukurtas kodas yra vykdomas ne naudotojo kompiuteryje o serveryje. Ši galimybė leidžia prisijungti prie išorinių šaltinių bei užtikrina aukštesnį saugumo lygį.



1 pav. Interneto svetainių kūrimo kalbos **PHP** logotipas [12]

Naudojantis **PHP.net** interneto svetainės kūrimo kalbos dokumentaciją [1] sužinota, kad iš viso **PHP** interneto svetainių kūrimo kalba turi 12 funkcijų:

- **HTTP autentifikavimas naudojant PHP.** Ši funkcija leidžia naudotis `header()` funkciją norint nusiųsti autentifikavimo reikalavimo žinutę kliento naršyklei;
- **slapukai.** **PHP** palaiko **HTTP** (angl. *Hypertext Transfer Protocol*) slapukus. Slapukai yra PHP mechanizmas skirtas saugoti duomenis nuotolinėse naršyklėse, stebėti arba identifikuoti šiuo kodu besinaudojančius svetainės naudotojus.
- **seansai.** **PHP** seansų palaikymas suteikia galimybę išsaugoti tam tikrus duomenis visuose vėlesniuose prieigose. Tai leidžia naudotojams sukurti skirtingas individualias aplikacijas norint pajvairinti svetaines;
- **XForma.** **XFormos** apibrėžia variacijas tradicinėse svetainės formose. Jos leidžia jas naudoti įvairesnėse platformose ir naršyklėse ar net netradicinėse laikmenose kaip **PDF** dokumentuose;
- **failų įkėlimai.** Ši funkcija leidžia naudotojams įkelti tekstą ir dvejetainius failus. Su **PHP** autentifikavimo ir failų modifikavimo funkcijomis svetainės kūrėjas gal kontroliuoti kas gali įkelti ir kas yra daroma su failu kai jis yra įkeltas;
- **nuotoliniai failai.** Galima naudotis **HTTP** ir **FTP** adresais su dauguma funkcijų, kurios nuskaito failo vardą kaip parametą. Tai leidžia valdyti failus nuotoliniuose serveriuose bei naudotis jų duomenimis;
- **ryšių palaikymas.** **PHP** turi keturis palaikomų ryšių statusai: normalus, nutrauktas, pertraukos ir nutrauktos pertraukos. Skriptui veikiant normalus statusas yra palaikomas. Jei nuotolinis klientas atsijungia, ryšio statusas pasikeičia į nutrauktą. Pertraukos statusas yra įjungiamas kai skripte yra nustatytas laiko terminas;

- **patvarios jungtys.** Šios jungtys yra nuorodos, kurios neužsidaro kai baigiasi skripto veikimas;
- **saugus režimas.** Saugos režimo tikslas yra išspręsti bendrų serverių apsaugos problemas. Ši funkcija naudoja papildomus saugos patikrinimus siekiant apsaugoti naudotojų aplinką;
- **PHP naudojimas komandinėje eilutėje.** **PHP** kalbą gali naudoti vykdyti skriptus komandinėse eilutėse;
- **šiukšlių rinkimas.** Ši funkcija tikrina kintamuosius ar jie yra naudojami ar ne. Vykdyt šiuokšlių rinkimo funkciją visi nenaudojami kintamieji yra pašalinami;
- **DTrace dinaminis sekimas.** **DTrace** gali sekti operacinių sistemų elgesį ir naudotojo programų vykdymą. Ši funkcija gali pavaizduoti argumentų vertę, kurios gali būti naudojamos sužinoti našumo statistikas.

Interneto svetainių kūrimo kalba **PHP** yra atviro kodo kalba. Tai reiškia kad **PHP** kodas yra atviras naudotojams ir organizacijoms norinčioms naudoti ir modifikuoti šią kalbą. **PHP 4**, **PHP 5** ir **PHP 7** yra platinami pagal **PHP** Licenciją v3.01.

Svarbiausias saugumo dalis yra naudotojo sugeneruota įvestis. Norint apsaugoti svetainės duomenis svarbu išsaugoti jos kritinę informaciją – naudotojų prisijungimo duomenis. Pagal elektroninę knygą *Comparative Studies of 10 Programming Languages within 10 Diverse Criteria* [10], kiekviena svetainių kūrimo kalba saugo informaciją nuo išorinių efektų skirtingai:

- **SQL** sąrašinių duomenų bazių užklausų kalbos injekcijos naudoja svetainės vietose, kurioje reikalingas konkreti reikšmė užklausoje į duomenų bazes. Prieš **SQL** injekcijas **PHP** naudoja **SQL** teiginius su **PHP** duomenų objektais;
- prieš **XSS** (angl. *Cross Site Scripting*), ši programavimo kalba naudoja funkcijas, pašalinančias pavojingas įvestis bei automatines šablonų sistemas;
- serverio kodo vykdymas naudojamas komandomis, galinčioms išvengti svetainės apsaugą. Tai leidžia vykdyti kodus išsaugotus kintamosiose. Vienintelis būdas tai išvengti yra nenaudoti šių funkcijų.

PHP programavimo kalba turi apsaugos priemonių prieš **SQL** injekcijas, **Cross-site** skriptavimą ir serverio kodo vykdymą, tačiau turi struktūros problemų, kurių rezultatas yra nepatikimas kalbos dizainas. **PHP** naudoja nesaugų programavimo struktūrą, kurios sukelia klaidas, susijusias su programos saugumu.

Interneto svetainių kūrimo kalba **PHP** palaiko įvairias platformas. Pagal oficialią dokumentaciją [1] **PHP** gali būti įrašoma į **Unix** operacinę sistemą. **Unix** – operacinė platforma

sukurta **UNICS** sistema, skirta **PDP** kompiuteriams. Kitos **PHP** palaikomos platformos yra **Mac OS**, **Windows**, **FPM** (angl. *FastCGI Process Manager*) ir debesų kompiuterinė platforma (angl. *Cloud*).

3.1.2 Interneto svetainių kūrimo kalba **Ruby**

Ruby programavimo kalba buvo sukurta 1995 metais. Jos kūrėjas, Yukihiro Matsumoto, sujungė programavimo kalbų **Perl**, **Smalltalk**, **Eiffel**, **Ada** ir **Lisp** elementus kad suformuotų naują kalbą, balansuojančią tarp funkcionalaus ir imperatyvaus programavimo [16]. Naudojant **Ruby** (žr. 2 pav.), kiekvienai kodo dalis tampa objektu. Objektinis programavimas iškviečia funkcijas ir veiksmus naudojantis klasėse apibrėžtais kintamaisiais, kitaip dar žinomus kaip metodais.



2 pav. Interneto svetainių kūrimo kalbos **Ruby** logotipas [16]

Ši kalba turi panašią sintaksę su kitomis programavimo kalbomis, kuri leidžia naudoti panašias ar visai naujas programavimo kalbų funkcijas. Iš viso **Ruby** interneto svetainės kūrimo kalba turi 15 funkcijų, jos yra pateikiama **Ruby** dokumentacijoje [15]:

- **iteracija.** **Ruby**, kitaip nei kitos programavimo kalbos kaip **Java** ir **PHP**, naudoja iteracijas blokus. Iteracijos yra programavimo metodai, kurie yra palaikomi kolekcijomis. Ši funkcija gražina vienas po kito visus kolekcijų elementus;
- **programavimo kodo vertė.** **Ruby** programavimo kalboje viskas turi specialią vertę. Išraiškos ir teiginiai programavimo kode neturi skirtumų. Viskas turi vertę, net jei ši vertė yra nulinė;
- **simboliai.** Simboliai gali būti apibūdinami kaip tapatybės. Kartą panaudojus simbolį, simbolis su tais pačiais charakteriais nurodo tą patį objektą atmintyje. Simboliai yra nekintantys, todėl simboliai yra vykdomi greičiau nei kintamieji. Tai leidžia greičiau vykdyti operacijas;

- **objektai.** Naudojant šią programavimo kalbą kiekviena klasė ir reikšmė yra objektai. Tai leidžia juos naudoti kaip kitą objektą. Ši funkcija leidžia kurti programas objektiniu programavimo būdu;
- **kintančios konstantos.** Nors *Ruby* programavimo kalboje konstantos turėtų išlikti pastovios, *Ruby* leidžia keisti konstantų reikšmes. Keičiant konstantą programavimo kalba perspėja kad konstantos reikšmė pasikeičia tačiau nestabdo naudojamos programos;
- **pavadinimų konvencijos.** *Ruby* įtvirtina dalį pavadinimų konvencijų. Jei identifikatorius prasideda su didžiąja raide, jis yra konstanta. Jei jis prasideda su dolerio ženklu (\$), jis yra globalus kintamasis. @ yra instancijos kintamasis, @@ - klasės kintamasis;
- **raktinių žodžių argumentai.** Nuo *Ruby 2.0* metodai gali būti apibrėžti kaip raktinių žodžių argumentai. Raktinių žodžių argumentai suteikia galimybę naudoti trumpinius su pakreiptaisiais brūkšneliais kaip argumentus metoduose;
- **reikšmės.** Programavimo kalboje *Ruby* visos reikšmės, išskyrus tas, kurios yra įvardytos kaip nulinės ir neteisingos, yra teisingos;
- **prieigos modifikatoriai.** Prieigos modifikatoriai yra taikomi iki srities pabaigos ar iki kito modifikatoriaus. Viešas, privatus ir apsaugotas metodai gali turėti parametrus. Jei metodui yra priskiriamas simbolis, metodo matomumas gali pasikeisti;
- **metodų prieinamumas.** *Ruby* metodų prieinamumas skiriasi nuo kitų programavimo kalbų. Viešas metodas yra prieinamas visoms kodo dalims, privatus metodas yra prieinamas tik kai jis yra vadinamas be aiškaus gavėjo. Tik asmeninis (angl. *self*) metodas gali būti privataus metodo gavėju;
- **klasės.** Visos klasės yra atviros, net objektai ir fiksuoto skaičiaus (angl. *Fixnum*) klasė. Jos gali būti atidaromos ir uždaromos bet kurio metu;
- **metodų pavadinimai.** *Ruby* kalboje metodai gali pasibaigti su klaustuko ar šauktuko ženklais. Taip galima surūšiuoti metodus naudotojui patogiu būdu;
- **pavieniai metodai.** Pavieniai metodai yra už objekto naudojami metodai. Šie metodai yra prieinami tik objektams, kuriuose jie yra apibrėžti;
- **trūkstanti metodai.** Nusiuntus žinutę objektui, objektas vykdo pirmą numatytą metodą su žinutės pavadinimu. Jeigu objektas neranda atitinkamo metodo, tada *Ruby* vykdo `method_missing` metodą su nerasto metodo pavadinimu. Šis metodas iškviečia `NameError` išimtį. Ši išimtis gali būti modifikuojama kad atitiktų naudojamą aplikaciją;

- **blokai.** Blokuose yra kodo dalys tarp `do` ir `end`, kurios gali bendradarbiauti su metodo inovacijomis. Blokas nėra iškart vykdomas, vietoj to **Ruby** įsimina bloko atsiradimo vietą ir tada įeina į metodą.

Kiekviena programavimo kalba kovoja prieš saugumą ir duomenis išskylančias grėsmes. Interneto svetainių saugumas yra aprašomas Klaus Purer elektroninėje knygoje **PHP vs. Python vs. Ruby – The web scripting language shootout** [14]. Kovoje prieš **SQL** injekcijas **Ruby** naudoja automatinę pagalbą, bet tik naudojant **Rails** struktūrą. Prieš **XSS** naudoja funkcijas, pašalinančias pavojingas įvestis arba automatines šablonų sistemas. Norint apsisaugoti nuo piktybiško serverio kodo vykdymo rekomenduojama nenaudoti saugumą apeinančias programų vykdymo komandas.

Ruby programavimo kalba yra palaikoma ant **Windows**, **Mac OS** ir **Unix** platformų [15]. **Ruby** galima įrašyti skirtingais būdais: naudojant paketų tvarkyklę, diegimo programą arba įrašymo vadovą. Taip pat šią programavimo kalbą galima įrašyti iš šaltinio failų.

3.1.3 Interneto svetainių kūrimo kalba **Python**

Python programavimo kalbą sukūrė Guido van Rossum 1985 – 1990 metais. **Python** (žr. **Error! Reference source not found.** pav.) Pagal **Python** dokumentaciją [13], **Python** interneto svetainės kūrimo kalba pasižymi programavimo kodo aiškumu ir sintaksę, leidžiančią išreikšti norimas koncepcijas naudojant mažiau kodo eilučių nei kitomis kalbomis kaip **C++** ar **Java**. Programavimo kalba buvo suprojektuota stengiantis neapkrauti programos vizualaus išdėstymo, naudojant anglų kalbos raktažodžius vietoj skyrybos ženklų.



3 pav. Interneto svetainių kūrimo kalbos **Python** logotipas [6]

Python programavimo kalba yra aukšto lygio interpretuojama interaktyvi objektais orientuota skriptavimo kalba, kurios pagrindiniai bruožai yra aiškumas ir suprantamumas. Pagal **Tutorial Point** internetinę svetainę [17] ir **Python** dokumentaciją [13], **Python** turi 13 funkcijų, kurios yra šios:

- **interaktyvus režimas.** **Python** palaiko interaktyvų režimą, kuris leidžia testuoti ir derinti atskiras kodo dalis;

- **portatyvus.** Ši svetainių kūrimo kalba gali veikti ant įvairių techninės įrangos platformų, išsaugojant tokią pačią naudotojo sąsają;
- **modulių didinimas.** Naudotojai gali įtraukti žemo lygio modulius į **Python** interpretatorių. Šie moduliai suteikia galimybę programuotojams pridėti ir keisti naudojamus programavimo kalbos įrankius;
- **duomenų bazės.** **Python** suteikia sąsajas su visomis pagrindinėmis komercinėmis duomenų bazėmis kaip **Oracle**, **Microsoft SQL** arba **PostgreSQL**;
- **GUI programavimas.** **Python** palaiko **GUI** aplikacijas, kurios gali būti sukurtos ir perkeltos į įvairias bibliotekas ir **Windows** sistemas kaip **Windows MFC**, **Macintosh** ir **X Windows** sistemą, kuri yra naudojama **Unix** programavimo platformoje;
- **struktūra.** **Python** programavimo kalba suteikia geresnę struktūrą ir paramą didelio tipo programoms nei **Shell** skriptavimas;
- **integravimas.** **Python** gali būti integruota į sukurta aplikaciją norint suteikti programuojamą sąsają;
- **pagrindiniai duomenų tipai.** Duomenų tipų įvairovė: skaičiai (slankaus kablelio, sudėtingi ir neriboto ilgio sveikieji skaičiai), eilutės tipai (**ASCII** ir **Unicode**), sąrašai ir žodynai;
- **objektinis programavimas.** **Python** palaiko objektinį programavimą naudojant klases ir paveldėjimais;
- **skirstymas.** Programoje sukurtas programos kodas gali būti sugrupuotas į modulius ir paketus;
- **išimtys.** Programavimo kalba palaiko įvairių tipo išimtis, taip palaikantis tvarkingą klaidų tvarkymą;
- **pažangios programavimo funkcijos.** **Python** palaiko pažangias programavimo funkcijas kaip generatoriai ir sąrašų supratimą;
- **atminties valdymas.** **Python** automatinis atminties valdymas atlaisvina naudotoją nuo būtinybės rankiniu būdu paskirsti ir atlaisvinti atminties rašomam kodui;

Python yra **FLOSS** (angl. *Free/Libre© and Open Source Software*) atviro kodo kalba. Programinė įranga yra laisvai platinama, galima skaityti programos kodą, jį koreguoti ir naudoti kodų dalis kituose programose. **Python** yra nuolat tobulinama šios kalbos bendruomene. Tai leidžia neribotą šios programinės kalbos keitimo galimybes.

Programavimo kalboje prieš **SQL** injekcijas **Python** kovoja naudojantis **API** duomenų bazę [14]. Siekiant apsaugoti nuo **XSS** yra naudojamos automatinės šablonų sistemos ir pavojų pašalinančias funkcijas. Siekiant apsaugoti nuo piktybiško serverio kodo vykdymo rekomenduojama nenaudoti saugumą apeinančias komandas.

Python programavimo kalbą galima naudoti įvairiuose platformose. **Python** turi dvi skirtingas versijas: **Python 2** ir **Python 3**. Pirmoji versija ir užbaigta ir nebeatnaujinama, antroji versija yra nuolat tobulinama [13]. Šią programavimo kalbą palaiko **Windows**, **Unix** ir **Mac OS X**.

3.1.4 Interneto svetainių kūrimo kalbų apžvalgos apibendrinimas

Ankstesniuose skyriuose buvo aptartos **PHP**, **Ruby** ir **Python** svetainių kūrimo kalbos, skirtingų programavimo kalbų funkcijos, kodų tipai, pasirinktos kalbos galimi saugumo pažeidimai ir palaikomos operacinės platformos. Išsiaiškinus prieš tai minėtus aspektus bus pateikiamas analizuotas interneto svetainių kalbų apibendrinimas pagal iškeltus aspektus. Programavimo kalbos yra lyginamos 8 lentelėje.

8 lentelė. Interneto svetainių kalbų palyginimas

Nr.	Kriterijus	PHP	Ruby	Python
1.	Funkcijų kiekis	12	15	13
2.	Kodo tipas	Atviras	Atviras	Atviras
3.	Saugumas:			
	SQL injekcija	Palaiko	-	Palaiko
	Cross-site skriptavimas	Palaiko	Palaiko	Palaiko
	Serverio kodo vykdymas	Palaiko	Palaiko	Palaiko
	Saugus kalbos dizainas	-	Palaiko	Palaiko
4.	OS palaikymas:			
	Unix	Palaiko	Palaiko	Palaiko
	Windows	Palaiko	Palaiko	Palaiko
	OS X	Palaiko	Palaiko	Palaiko
	Cloud	Palaiko	Palaiko	Palaiko
	FPM	Palaiko	-	-

FPM nėra operacinė sistema, ji yra **FastCGI** procesų valdiklis, naudojamas atlikti **FastCGI PHP** svetainės kalboje.

Iš 8 lentelės matyti kad funkcijų kiekis nežymiai svyruoja tarp skirtingų programavimo kalbų. Pastebėta, kad **Ruby** programavimo kalba turi daugiausiai skirtingų funkcijų.

Visos trys interneto svetainių kūrimo kalbos yra atviro kodo, tačiau dalis **PHP** failų yra platinami pagal kitas licencijas, kurioms reikia papildomų reikalavimų ir autorių teisių informacijos.

Saugumo aspektu pastebima, kad visos trys interneto svetainių programavimo kalbos yra apsaugotos nuo kodo skriptavimo (angl. *cross-site scripting*) ir nelegalaus serverio kodo vykdymo. Kovoiant prieš **SQL** injekcijas pastebėta, kad **Ruby** programavimo kalba yra apsaugota naudojant **Rails** struktūrą. Tačiau **PHP** turi prastą programavimo kalbos dizainą, sukelia klaidas, susijusias su programos saugumu.

Visos interneto svetainių kūrimo kalbos teikia paslaugas **Unix**, **Windows**, **OS X** ir **Cloud** sistemoms, tačiau **PHP** taip pat teikia paslaugas ir **FPM** procesų valdikliui.

Pagal iškeltus kriterijus aptarus interneto svetainių kūrimo kalbas nustatyta, kad **Ruby** turi daugiausia funkcijų – 15. Antroje vietoje **Python** suteikia naudotojams 13 funkcijų o trečioje vietoje **PHP** – 12. Visos trys programavimo kalbos yra atviro kodo. Saugumo aspektu, **PHP** naudoja nesaugu kalbos dizainą, kuris suteikia galimybę naudotojams sukurti saugumo spragas. Nuo **SQL** injekcijų, **Cross-site** skriptavimo ir nelegalaus serverio kodo vykdymo visos trys kalbos yra apsaugotos. Interneto svetainių kūrimo kalba **PHP**, **Ruby** ir **Python** teikia paslaugas operacinėms sistemoms **Unix**, **Windows**, **OS X** ir **Cloud**, tačiau **FPM** sistemai teikia paslaugas tik **PHP**.

3.2. Interneto svetainių kūrimo kalbų analizės atlikimo kriterijai ir vertinimo skalės

Prieš atliekant interneto svetainių kūrimo kalbų palyginimo analizę ir bandymus, reikalinga apibrėžti naudojamus kriterijus. Jais naudojantis atliekamas interneto svetainių kūrimo kalbų **PHP**, **Ruby** ir **Python** palyginimai ir bandymai. Siekiant išsamiai išanalizuoti kiekviena svetainės kūrimo kalbą naudojantis **PHP**, **Ruby** ir **Python** sukuriamos trys vienodos interneto svetainės. Šių kalbų vertinimo kriterijai yra lygiaverčiai ir kiekvienas iš jų sudaro 20% vertinimo balų. Kriterijai yra pateikti 9 lentelėje.

9 lentelė. Interneto svetainių kūrimo kalbų kriterijai ir įvertinimai

Nr.	Kriterijus	Pagal ką vertinimas	Įvertinimas
1	Svetainės kalbos interpretatoriaus parsisiuntimas ir įdiegimas	Pavyko parsisiųsti ir įdiegti	2
		Pavyko parsisiųsti bet nepavyko įdiegti	1
2	Svetainės apdorojamų užklausų kiekis	Apdoroja daugiausiai naudotojų užklausų per sekundę	3
		Apdoroja vidutinį kiekį naudotojų užklausų per sekundę	2
		Apdoroja mažiausiai naudotojų užklausų per sekundę	1
3	Duomenų bazės poveikis svetainės naudotojų užklausų vykdymui	Mažiausiai pakito apdorojimo kiekio efektyvumas	3
		Vidutiniškai pakito apdorojimo kiekio efektyvumas	2
		Daugiausiai pakito apdorojimo kiekio efektyvumas	1
4	Svetainės apdorojamų užklausų poveikis procesoriams	Naudoja mažiausiai procesoriaus galios	3
		Naudoja vidutiniškai procesoriaus galios	2
		Naudoja daugiausiai procesoriaus galios	1
5	Svetainės kodo eilučių skaičius	Programa, kuri turi mažiausiai kodo eilučių skaičių	3

Nr.	Kriterijus	Pagal ką vertinimas	Įvertinimas
		Programa, kuri turi vidutinį kodo eilučių skaičių	2
		Programa, kuri turi daugiausia kodo eilučių skaičių	1

Svetainės kalbos interpretatoriaus parsisiuntimas ir įdiegimas. Lyginama, kaip sklandžiai pavyko parsisiųsti ir įdiegti interneto svetainių kūrimo kalbų *PHP*, *Ruby*, *Python* interpretatorius bei jų atitinkamus serverius ir sąsajas. Aiškinamasi ar parsisiuntimo ir įrašymo eiga vyko sklandžiai. Rezultatai yra vertinami naudojantis 9 lentele. Jei procesas įvyko sklandžiai ir be netikėtų kliūčių parsisiuntimui ir įdiegimui skiriami 2 balai, pavykus parsisiųsti tačiau iškilus problemų įrašant programavimo kalbą kriterijui yra skiriamas 1 balas.

Svetainės apdorojamų užklausų kiekis. Tikrinama, kiek naudotojų užklausų per sekundę gali palaikyti kiekvienos interneto svetainės kūrimo kalbos svetainė. Balai skiriami didėjimo tvarka. Mažiausiai apdorojusi programavimo kalba įvertinama vienu balu, antroje vietoje esanti programavimo kalba įvertinama dviem balais, o daugiausia užklausų įvykdžiusi kalba – trimis balais.

Duomenų bazės poveikis svetainės naudotojų užklausų vykdymui. Tikrinama, kaip *MySQL* duomenų bazių valdymo sistemoje esanti duomenų bazės `items` dydis įtakoja svetainės užklausų apdorojimo kiekį. Įvykdomi trys etapai, kuriuose testuojama su skirtingo duomenų bazės eilučių skaičiumi – viena eilute, penkiasdešimt eilučių ir šimtas eilučių. Palyginami kiekvienos interneto svetainės kūrimo kalbos apdorojimo efektyvumo pokytis procentais. Programavimo kalbos, kurios apdorojimo kiekis mažiausiai pasikeitė yra įvertinama trimis balais, antroje pozicijoje esanti kalba įvertinama dviem balais ir daugiausiai pasikeitęs kalbos efektyvumas įvertinamas vienu balu.

Svetainės apdorojamų užklausų poveikis procesoriams. Tikrinamas svetainės apdorojamų užklausų poveikis nešiojamųjų kompiuterių procesoriams. Vykdamant naudotojų užklauses yra tikrinamas nešiojamųjų kompiuterių procesorių padėtis. Mažiausiai procesoriaus galios naudojanti kalba yra vertinama trimis balais, antroje vietoje esanti kalba vertinama dviem balais, o kalba naudojanti daugiausiai procesoriaus galios – vienu balu.

Svetainės kodo eilučių skaičius. Sukūrus svetainę yra suskaičiuojamas panaudotų eilučių skaičius. Siekiant tiksliai apskaičiuoti eilučių skaičiui naudojama *CLOC* (angl. *Count Lines of Code*) programa. Jei programavimo kalbos kodas turi mažiausią eilučių skaičių, jai yra skiriami trys balai. Didžiausią kodo eilučių skaičių turinti programa yra vertinama vienu balu. Antroje pozicijoje esanti kalba yra vertinama dviem balais.

Šiame poskyryje aptariamos **PHP**, **Python** ir **Ruby** svetainės kūrimo kalbos bei atliekami bandymai siekiant nustatyti programavimo kalbų skirtumus. Norint sužinoti esamus skirtumus yra atliekami šie bandymai: parsiumčiama ir įdiegiama programavimo kalbos interpretatorius, tikrinamas kiekvienos svetainės apdorojamų užklausų kiekis, testuojamas *items* duomenų bazės poveikis svetainės naudotojų užklausų vykdymui, apžvelgiamas svetainės apdorojamų užklausų poveikis procesoriams bei patikrinamas interneto svetainių kūrimo kalbų svetainių kodo eilučių skaičius.

Sklandžiai įvykdžius interneto svetainės kūrimo kalbos interpretatoriaus parsiumtimą ir įdiegimą, kriterijus yra įvertinamas dviem balais. Iškilus problemoms įrašant programavimo kalbos interpretatorių įvertinimas sumažinamas iki vieno balo. Svetainės apdorojamų užklausų kiekis, duomenų bazės poveikis svetainės užklausų vykdymui, svetainės užklausų poveikis procesoriams ir svetainės kodo eilučių skaičius yra vertinamas trijų balų sistema.

3.3. Interneto svetainių kūrimo kalbų palyginamosios analizės bandymų atlikimo aplinka

Norint atlikti interneto svetainių kūrimo kalbų palyginamąją analizę reikia išsiaiškinti analizės atlikimo bandymų aplinką. Bandymuose yra naudojami nešiojamieji kompiuteriai. Bandymo priemonių specifikacijos pateikiamos 10 lentelėje.

10 lentelė. Bandymo priemonių apžvalga

Nr.	Pavadinimas	Operacinė sistema	Kietasis diskas	Procesorius	Operatyvioji atmintis	Atmintis
Nešiojamieji kompiuteriai						
1.	Samsung NP350V5C	Windows 10 Pro	ST500LM012 HN	Intel Core i3-3110M	4 GB	500 GB
2.	Lenovo g510	Windows 8.1	ST1000LM024 HN-M101-MBB 931GB	Intel Core i3-4000M	4 GB	500 GB
3.	HP 15-n245sa	Windows 7 Home	Toshiba MQ01ABD075	Intel Core i5-4200U	6 GB	750 GB

Iš lentelės matyti kad bandyme bus naudojami trys nešiojamieji kompiuteriai: **Samsung NP350V5C**, **Lenovo g510** ir **HP 15-n245sa** ir naudoja **Windows** operacines sistemas. Bandymuose naudojamų operacinių sistemų versijos, procesoriai, operatyvioji atmintis bei atmintis skiriasi.

Interneto svetainių kūrimo kalbos naudoja skirtingų tipų failų formatus. Jie yra išvardijami sąrašo forma sekančioje eilutėje:

- **PHP** failo formatas;
- **Python** failo formatas;

- **Ruby** failo formatas;
- **Rack** failo formatas;
- **CSS** failo formatas;
- **HTML.ERB** failo formatas.

Bandymuose yra naudojamos skirtingų tipų programinės įrangos. Naudojami techninė įranga pateikiama sąrašo forma:

- **Notebook++** programavimo kodo redaktorius. Šia programa programuojamas svetainės kodas;
- **Wamp** serveris. Naudojamas kuriant interneto svetainę su **PHP** ir **Python** programavimo kalba;
- **Rack** serverio sąsaja. Palaiko minimalią sąsają tarp **Ruby** interneto serverių. Naudojamas kuriant interneto svetainę su **Ruby** programavimo kalba;
- **Chrome** žiniatinklio naršyklė;
- **Command Prompt** kodo eilučių interpretatorius. Naudojamas vykdyti **Ruby** interneto svetainės įvestas komandas;
- **MySQL** duomenų bazių valdymo sistema. Valdymo sistemoje yra naudojama duomenų bazė `items`, kuri jungiasi su kiekviena interneto svetaine;
- **Task Manager** įrankis. Naudojantis šiuo įrankiu yra pateikia informacija apie kompiuteryje vykdomus procesus bei vaizduojama sistemos informaciją;
- **Cloc** kodo eilučių skaičiavimo programa. Kartu su **Command prompt** kodo eilučių interpretatoriumi naudojantis šia programa galima gauti informaciją apie interneto svetainėje naudojamų kalbų kodo eilučių skaičių.

Naudojantis skirtingų tipų failų formatais ir programines įrangas yra kuriamos **PHP**, **Python** ir **Ruby** interneto svetainės.

3.4. Interneto svetainių kūrimo kalbų palyginamosios analizės bandymų procesai

Išsiaiškinus bandymuose naudojamus kriterijus, kriterijų vertinimo skales ir procesų atlikimo aplinką sudaromas bandymų atlikimo eiga. Kad bandymai būtų sėkmingai yra sukuriamas planas, pagal kurį kriterijai yra patikrinami atlikimo aplinkose, naudojamomis skirtingomis interneto svetainių kūrimo kalbomis. Sudarytas bandymo planas yra pavaizduotas 11 lentelėje.

11 lentelė. Interneto svetainių kūrimo kalbų bandymų atlikimo matrica

Nr.	Kriterijai	Priemonės	Bandymai		
			PHP	Python	Ruby
1 etapas					
1.	Svetainės kalbos interpretatoriaus parsisiuntimas ir įdiegimas	Samsung NP350V5C	1	4	7
		Lenovo g510	2	5	8
		HP 15-n245sa	3	6	9
2 etapas					
2	Svetainės apdorojamų užklausų kiekis	Samsung NP350V5C	10	13	16
		Lenovo g510	11	14	17
		HP 15-n245sa	12	15	18
3	Duomenų bazės poveikis svetainės naudotojų užklausų vykdymui	Samsung NP350V5C	19	22	25
		Lenovo g510	20	23	26
		HP 15-n245sa	21	24	27
4	Svetainės apdorojamų užklausų poveikis procesoriams	Samsung NP350V5C	28	31	34
		Lenovo g510	29	32	35
		HP 15-n245sa	30	33	36
5	Svetainės kodo eilučių skaičius	Samsung NP350V5C	37	38	39

Pagal 11 lentelės duomenis matyti, kad interneto svetainių kūrimo kalbų bandymo atlikimo matrica susideda iš dviejų etapų.

1 etapas. Pirmu etapu yra parsisiunčiamos ir įdiegiamos interneto svetainių kūrimo kalbų interpretatoriai ir jų serverius bei sąsajas į prieš tai išvardytus nešiojamuosius kompiuterius. Su kiekviena analizuojama programavimo kalba yra atliekami 3 bandymai. Šiame etape bendrai yra atliekami 9 bandymai.

2 etapas. Antrame etape testuojamos interneto svetainės kūrimo kalbos sukurtos svetainių apdorojamų užklausų kiekis, duomenų bazės poveikis svetainės naudotojų užklausų vykdymui, svetainės užklausų poveikis procesoriams ir svetainės kodo eilučių skaičius. Išskyrus svetainės kodo eilučių skaičių, visi kriterijai yra atliekami devyniais bandymais, trejais bandymais kiekvienai kalbai. Svetainės kodo eilučių skaičius yra tikrinamas trimis bandymais, naudojantis **Samsung NP350V5C** nešiojamuoju kompiuteriu. Iš viso šiame etape yra atliekami 30 bandymų, nuo 10 iki 39 bandymo.

Apibendrinant šį poskyri išsiaiškinta interneto svetainių kūrimo kalbų bandymų atlikimo matricos struktūra. Ši struktūra yra išskaidyta į du etapus. Pirmame etape yra parsisiunčiamos ir įdiegiamos interneto svetainių kūrimo kalbos. Antrame etape yra tikrinimas svetainės apdorojamų užklausų kiekis, duomenų bazės poveikis svetainės naudotojų užklausų vykdymui, svetainės užklausų poveikis procesoriams ir svetainės kodo eilučių skaičius. Iš viso bandymo atlikimo matricą sudaro 39 bandymai.

3.5. Skyriaus apibendrinimas

Parengus interneto svetainių kūrimo kalbų palyginamosios analizės atlikimo metodiką nustatyta, kad daugiausia skirtingų funkcijų palaiko **Ruby** programavimo kalba, mažiausiai – **PHP**. Nustatyta kad visos trys programavimo kalbos naudoja atviro kodo tipą. Tikrinant interneto svetainių kalbų funkcijų kiekį pastebėta, kad visos programavimo kalbos turi panašų funkcijų skaičių, **Ruby** programavimo kalbai pirmaujanti pagal jų skaičių. Peržiūrint programavimo kalbų saugumą pastebėta, kad programavimo kalba **PHP** nepasižymi saugiu kalbos dizainu, o **Ruby** programavimo kalba nėra saugi nuo **SQL** injekcijų. **Python** programavimo kalba įvykdė visus saugumo kriterijus. Tikrinant palaikomas operacines pastebėta, kad **PHP**, **Ruby** ir **Python** palaiko **Unix**, **Windows**, **OS X** ir **Cloud** operacines sistemas, bet tik **PHP** palaiko **FPM** procesų valdiklį.

Aprašomi interneto svetainių kūrimo kalbų kriterijai ir vertinimo skalės. Programavimo kalbos vertinamos pagal iškeltus kriterijus: svetainės kalbos interpretatoriaus parsisiuntimas ir įdiegimas, svetainės apdorojamų užklausų kiekis, duomenų bazės poveikis svetainės naudotojų užklausų vykdymui, svetainės apdorojamų užklausų poveikis procesoriams ir svetainės kodo eilučių skaičius.

Visi bandymai yra atliekami naudojant tris skirtingus nešiojamuosius kompiuterius – **Samsung NP350V5C**, **Lenovo g510** ir **HP 15-n245sa**. Šie nešiojamieji kompiuteriai naudoja **Windows 10 Pro**, **Windows 8.1** ir **Windows 7 Home** operatyvias sistemas.

Bandymo procesas susideda iš dviejų etapų. Pirma etape parsisiunčiamos ir įdiegiamos interneto svetainių kalbų interpretatoriai. Antrame etape testuojamas interneto svetainės kūrimo kalbos svetainės: svetainės apdorojamų užklausų kiekis, duomenų bazės poveikis svetainės naudotojų užklausų vykdymui, svetainės apdorojamų užklausų poveikis procesoriams ir svetainės kodo eilučių skaičius.

4. INTERNETO SVETAINIŲ KŪRIMO KALBŲ PALYGINAMOSIOS ANALIZĖS ATLIKIMO EIGA, REZULTATŲ APIBENDRINIMAS IR APTARIMAS

Praeitame skyriuje buvo aptarta interneto svetainių kūrimo kalbų *PHP*, *Python* ir *Ruby* palyginamosios analizės atlikimo metodika, kurioje buvo palygintos programavimo kalbos, apibrėžti bandymų kriterijai ir jų vertinimo kriterijai, nustatyta bandymų atlikimo aplinka ir jų procesai. Šiame skyriuje kalbama apie:

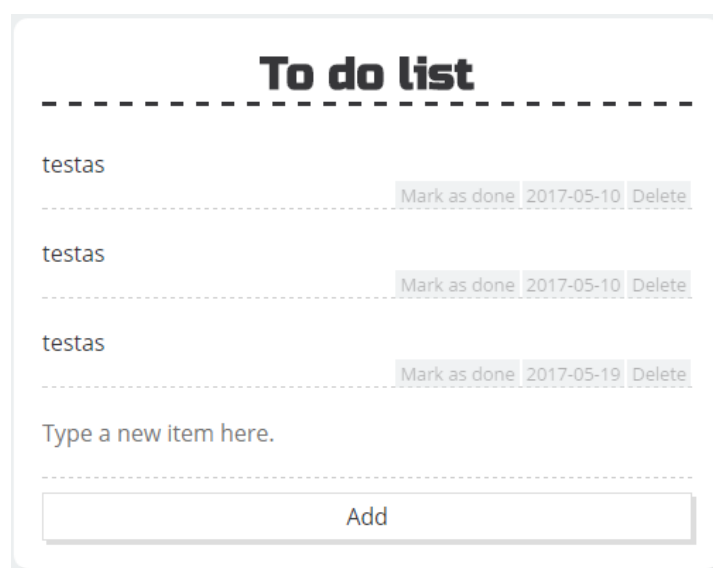
1. interneto svetainių kūrimo kalbų palyginamosios analizės eiga;
2. interneto svetainių kūrimo kalbų palyginamosios analizės bandymų rezultatai.

Naudojantis iškeltais programavimo kalbų kriterijais, testuojamos interneto svetainių kūrimo kalbos ir stiebiami rezultatai.

4.1. Interneto svetainių kūrimo kalbų palyginamosios analizės atlikimo eiga

Šiame skyriuje yra atliekama palyginamoji analizė, analizuojamos ir bandomos interneto svetainių kūrimo kalbos *PHP*, *Python* ir *Ruby* interpretatoriaus parsisiuntimas bei įdiegimas, vartotojų apdorojamų užklausų kiekis, duomenų bazės poveikis svetainės naudotojų užklausų vykdymui, svetainė apdorojamų užklausų poveikis procesoriui ir svetainės kodo eilučių skaičius.

Naudojantis interneto svetainės kūrimo kalbomis yra sukurama linijinės dinaminės interneto svetainės. Svetainė yra pavaizduojama 4 paveikslėlyje.

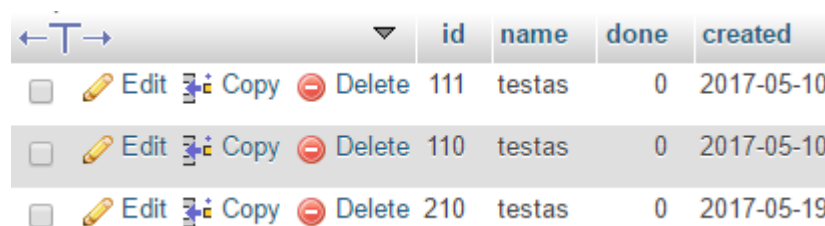


The screenshot shows a web application titled "To do list". It features a list of three items, each labeled "testas". To the right of each item are three buttons: "Mark as done", a date (2017-05-10 for the first two items and 2017-05-19 for the third), and "Delete". Below the list is a text input field with the placeholder "Type a new item here." and an "Add" button.

4 pav. Interneto svetainės struktūra

Svetainių paskirtis yra sudaryti darbų sąrašą. Svetainė yra sudarytos iš 5 pagrindinių detalių: teksto įvedimo lauko ir keturių mygtukų: teksto lauko užbraukimo, datos vaizdavimo, ištrynimo ir duomenų pridėjimo mygtuko. Įvedus tekstą į teksto lauką ir paspaudus duomenų pridėjimo mygtuką, duomenys yra išsaugojami į nemokamą **MySQL** duomenų bazių valdymo sistemos [9] `items` duomenų bazės lentelėje. Panaudojus užbraukimo mygtuką pasirinktos eilutės tekstas yra užbraukiamas bei užbraukimo ir datos mygtukai yra panaikinami. Datos mygtukas vaizduoja eilutės teksto sukūrimo datą, o ištrynimo mygtukas panaikintą sukurtą eilutę bei su ja susijusius mygtukus.

Duomenų bazės lentelė `items` yra sudaryta iš keturių stulpelių: `ID`, `name`, `done` ir `created`. Lentelės struktūra vaizduojama 5 paveikslėlyje.



				id	name	done	created
☐	Edit	Copy	Delete	111	testas	0	2017-05-10
☐	Edit	Copy	Delete	110	testas	0	2017-05-10
☐	Edit	Copy	Delete	210	testas	0	2017-05-19

5 pav. Duomenų bazės struktūra

`ID` yra duomenų lentelės identifikavimo numeris, `name` yra duomenų bazėje išsaugojamas tekstas, `done` naudojamas tikrinant ar sąrašo elementas yra pabaigtas ir `created` išsaugo sąrašo elemento sukūrimo datą. Pirmasis ir antrasis stulpelis saugo informaciją skaičiaus tipu, trečiasis yra tekstinis tipas, o ketvirtasis naudoja datos tipą, kuris išsaugo datą bei tikslų dienos laiką.

Naudojantis interneto svetaine ir su ja sujungta duomenų baze programavimo kalbos rezultatai yra aptariama atskirai eiliškumo tvarka.

4.1.1 Interneto svetainės kūrimo kalbos **PHP** bandymai

Svetainės kalbos interpretatoriaus parsisiuntimas ir įdiegimas. Programavimo kalbos **PHP** interpretatorius yra parsisiunčiamas į nešiojamuosius kompiuterius **Samsung NP350V5C**, **Lenovo g510** ir **HP 15-n245sa**. Norint pasiekti **PHP** programavimo kalbos interpretatoriaus ir **Wamp** serverio suderinamumą, programavimo kalbos interpretatorius yra parsisiunčiamas iš oficialios **Wamp** svetainės [21]. Paspaudus parsisiuntimo mygtuką, naudotojas yra nukreipiamas į skiltį, kurioje yra pateikiamos skirtingos **Wamp** versijos, suskirstytos į 32 ir 64 bitų. Parsiunčiama 3.0.6 versija.

Parsiuntus **Wamp** serverio paketą jis yra įdiegiamas. Aktyvavus serverio paketą yra įdiegiamos 5.6.25 **PHP** programavimo kalbos interpretatoriaus versija, 2.4.23 **Apache** programinė įranga, 5.7.14 **MySQL** duomenų bazės valdymo sistema bei kiti reikalingi įskiepijai. Įdiegimo

pabaigoje pasirenkama žiniatinklio naršyklė **Chrome**, kuri bus naudojama svetainės vaizdavimui. Programavimo kalbos interpretatoriaus parsisiuntimas ir įdiegimas vyko sklandžiai.

Svetainės apdorojamų užklausų kiekis. Bandymų metu yra tikrinamas sukurtos programavimo kalbos svetainės apdorojamų užklausų skaičius per vieną sekundę. Tam apskaičiuoti yra naudojamas **Apache** įrankis **Apache Bench (ab)**. Įrankis yra naudojamas **Command prompt** kodo interpretatoriumi.

Šis įrankis yra naudojamas komanda **ab -n 100 -c 10 -k -H "Accept-Encoding: gzip, deflate"**. **-n** tekstas ir sekantis skaičius nusako kiek užklausų yra įvykdoma. Vykdomų užklausų skaičių vykdomų vienu metu nurodo **-c** tekstas ir sekantis skaičius. **-k** komanda nurodo, kad pasibaigus vienai užklausiai pradedama sekanti užklausa. Paskutinė komandos dalis tikrina ar yra kompresavimo metodų, suradus metodą jis yra aktyvuojamas.

Siekiant tiksliai apskaičiuoti vidutinį svetainės apdorojamų užklausų skaičių per sekundę, kiekvienam nešiojamajam kompiuteriui yra atliekami 50 testų. Testų rezultatų vidurkiai yra sudedami ir padalijami iš įvykdytų testų aplinkų skaičiaus. Testo rezultatai yra pateikiami 12 lentelėje.

12 lentelė. **PHP** svetainės užklausų vidurkis vykdant 100 užklausų

Programavimo kalba	<i>Samsung NP350V5C</i> svetainės užklausų vidurkis per 1 sekundę	<i>Lenovo g510</i> svetainės užklausų vidurkis per 1 sekundę	<i>HP 15-n245sa</i> svetainės užklausų vidurkis per 1 sekundę
PHP	598,19	605,16	852,57

Išbandžius **PHP** interneto svetainės kūrimo kalbos svetainę su skirtingu nešiojamųjų kompiuteriu išsiaiškinta, kad nešiojamasis kompiuteris **Samsung NP350V5C** vidutinis svetainės užklausų apdorojimo kiekis per vieną sekundę yra 598,19 užklausų. **Lenovo g510** vidutiniškai apdoroja 605,16 svetainės užklausų per sekundę, o **HP 15-n245sa** – 852,57 svetainės užklausas per sekundę.

Duomenų bazės poveikis svetainės naudotojų užklausų vykdymui. Šiuo bandymu yra tikriname, kaip **items** duomenų bazės dydis įtakoja svetainės užklausų apdorojimo kiekį. Interneto svetainės kūrimo kalbos svetainė testuojama trejose stadijose: duomenų bazėje esant vienai eilutei, eilučių skaičius esant 50 ir kai eilučių skaičius yra lygus 100. Kiekvienai stadijai įgyvendinti yra vykdomi 20 testų, iš kurių išvedamas vidutinis svetainės užklaustos laikas.

Kaip ir praeituose užklausų vykdyme naudosime **Apache** įrankį **Apache Bench. Command prompt** kodo interpretatoriuje yra įvedama komanda **ab -n 100 -c 10 -k -H "Accept-Encoding: gzip, deflate"**. Testo rezultatai yra pateikiami 13 lentelėje.

13 lentelė. Duomenų bazės poveikis **PHP** svetainės užklausų vykdymui

Duomenų bazės eilučių skaičius PHP svetainėje	Samsung NP350V5C svetainės užklausų vidurkis per 1 sekundę	Lenovo g510 svetainės užklausų vidurkis per 1 sekundę	HP 15-n245sa svetainės užklausų vidurkis per 1 sekundę
1 eilutė	595,76	613,72	856,36
50 eilučių	413,50	443,47	561,70
100 eilučių	371,17	406,49	515,43

Lentelėje pavaizduoti kiekvieno nešiojamojo kompiuterio vidutinis užklausų apdorojimo kiekis per 1 sekundę, kai duomenų bazėje yra 1 eilutė, 50 eilučių ir 100 eilučių. Iš šios lentelės mes galime sugeneruoti 14 lentelę.

14 lentelė. Procentinis duomenų bazės poveikis **PHP** svetainės užklausų vykdymui

Duomenų bazės eilučių skaičius PHP svetainėje	Samsung NP350V5C svetainės užklausų apdorojimo kiekis, %	Lenovo g510 svetainės užklausų apdorojimo kiekis, %	HP 15-n245sa svetainės užklausų apdorojimo kiekis, %
1 eilutė	100 %	100 %	100 %
50 eilučių	69,41 %	72,26 %	65,59 %
100 eilučių	62,30 %	66,23 %	60,19 %

Apdorojimo kiekis, bandant duomenų bazę su viena įvestimi yra laikomas maksimaliu programavimo kalbos apdorojimo kiekiu. Padidinus duomenų bazės eilučių skaičių iki 50, **Samsung**, **Lenovo** ir **Hp** kompiuterių apdorojimo efektyvumai sumažėjo: **Samsung** apdorojimo kiekis sumažėjo iki 69,41 %, **Lenovo** sumažėjo iki 72,26 %, o **Hp** – 65,59 %. Padidinus duomenų bazės eilučių skaičių iki 100, atitinkamai nešiojamųjų kompiuterių apdorojimo efektyvumas sumažėjo iki 62,30 %, 66,23 % ir 60,19 %.

Svetainės apdorojamų užklausų poveikis procesoriui. Tikrinamas svetainės apdorojamų užklausų poveikis nešiojamųjų kompiuterių procesoriams. Šiam bandymui įvykdyti yra naudojamas **Apache** įrankis **ab** bei **Task Manager**. Įrankis **ab** yra vykdomas **Command prompt** kodo interpretatoriuje, naudojantis komanda `ab -n 100 -c 10 -k -H "Accept-Encoding: gzip, deflate"`. Procesoriaus užimtumas yra tikrinamas **Task Manager** programoje.

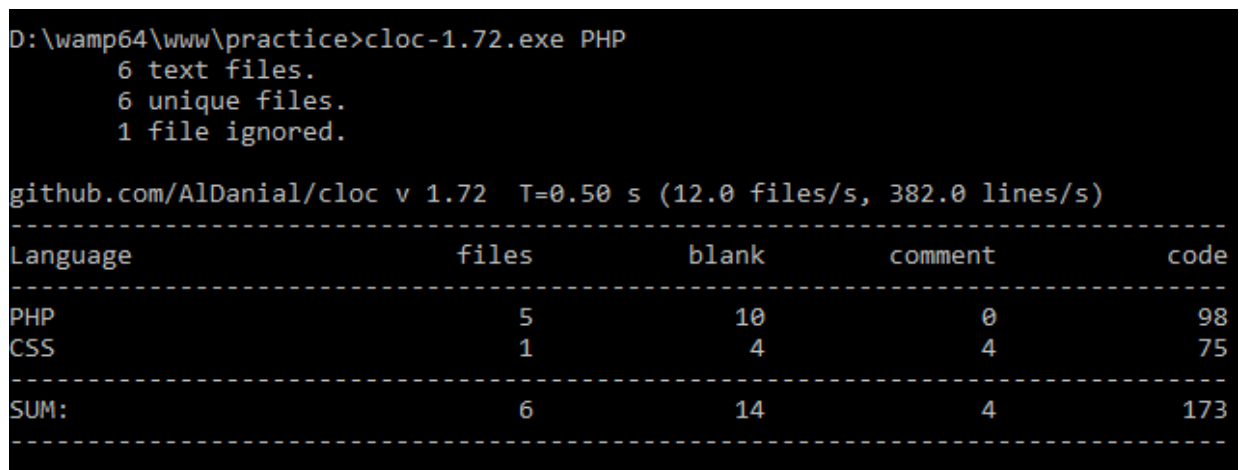
Kad surinkti duomenys būtų tikslesni, svetainės užklausų poveikis procesoriams yra tikrinamas atliekant 50 testų skirtinguose nešiojamuosiuose kompiuteriuose. Skirtingų nešiojamųjų kompiuterių rezultatų vidurkiai yra sudedami ir padalijami iš naudojamų testų aplinkų skaičiaus. Testo rezultatai yra pateikiami 15 lentelėje.

15 lentelė. **PHP** svetainės apdorojamų užklausų poveikis procesoriui vykdant 100 užklausų

Programavimo kalba	Samsung NP350V5C procesoriaus galios sunaudojimas, %	Lenovo g510 procesoriaus galios sunaudojimas, %	HP 15-n245sa procesoriaus galios sunaudojimas, %
PHP	19,88 %	18,58 %	16,36 %

Išbandžius **PHP** interneto svetainės kūrimo kalbos svetainę ant skirtingų nešiojamųjų kompiuterių išsiaiškinta, kiek procentų procesoriaus galios naudoja įvykdant 100 svetainės užklausų. **Samsung** nešiojamasis kompiuteris vidutiniškai naudoja 19,88 % procesoriaus galios, **Lenovo** naudoja 18,58 % galios o **Hp** nešiojamasis kompiuteris – 16,36 % procesoriaus galios. Procesorių galios sunaudojimų vidurkis yra 18,27 %.

Svetainės kodo eilučių skaičius. **PHP** svetainės kodo eilučių skaičius yra tikrinamas naudojantis programa **Cloc**. Ši programa yra parsisiunčiama iš programos publikuotos svetainės [5]. Parsiuntus programą, yra įjungiamas **Command prompt** interpretatorius. Naudojantis šia programa yra aktyvuojama **Cloc** programa nurodant programos pavadinimą ir norimą failo pavadinimą, esantį toje pačioje alokacijoje. Programos rezultatas yra pateikiamas 6 paveikslėlyje.



```
D:\wamp64\www\practice>cloc-1.72.exe PHP
6 text files.
6 unique files.
1 file ignored.

github.com/AlDanial/cloc v 1.72  T=0.50 s (12.0 files/s, 382.0 lines/s)
-----
Language             files            blank           comment           code
-----
PHP                   5                10                0                98
CSS                   1                 4                 4                75
-----
SUM:                  6                14                 4               173
-----
```

6 pav. **PHP** programavimo kalbos svetainės kodo tipai ir eilučių skaičius

Iš paveikslėlio matyti failo informacija yra suskirstyti į penkias dalis: naudojama programavimo kalba, failų skaičius, tuščios kodo eilutės, komentarai ir kodo eilutės. Programavimo kalba nurodo kokios programavimo kalbos yra naudojamos nurodytame aplanke. Programoje naudojamos **HTML** svetainių kūrimo kalba yra įskaičiuojama į **PHP** eilučių skaičių. Tuščios kodo eilutės nurodo kiek yra palikta tuščių kodo eilučių. Komentarai nurodo autoriaus parašytus komentarus, kurie neįtakoja programos veikimo. Paskutinė skiltis nurodo kiekvienos kalbos kodo eilučių skaičių. Tuščios eilutės, autoriaus komentarai ir kodo eilučių skaičius yra susumuojamos ties **SUM** eilute.

Siekiant išsiaiškinti kodo eilučių skaičių, iš bendro kodo eilučių kiekio yra atimamas komentarų ir tuščių kodo eilučių skaičius. Kadangi **CSS** kodo eilučių skaičius yra vienodas skirtingų kalbų programose, eilučių skaičius yra įskaičiuojamas į bendrą kodo eilučių skaičių. **PHP** programavimo kalba naudoja 155 kodo eilutes.

4.1.2 Interneto svetainės kūrimo kalbos *Python* bandymai

Svetainės kalbos interpretatoriaus parsisiuntimas ir įdiegimas. Norint parsisiųsti *Python* programavimo kalbos interpretatorių, yra aplankoma programavimo kalbos oficialus interneto svetainė <https://www.python.org>. Svetainėje aktyvavus parsisiuntimo mygtuką naujame tinklapyje yra pasirenkama 3.5.0 interneto svetainės kūrimo kalbos interpretatoriaus versija. Pasirenkama naudojama operacinė sistema ir parsisiunčiamas failų paketas.

Parsiųstas failų paketas yra aktyvuojamas ir pradedamas instaliavimo procesas, kuriame pasirenkamas programavimo kalbos interpretatoriaus įrašymo vieta. Pasibaigus įrašymo procesui yra koreguojamas Apache `httpd.conf` failas, kuriame yra pridamos kodo eilutės `AddHandler cgi-script .cgi .py` ir `Options Indexes FollowSymLinks ExerCGI` siekiant naudoti *Python* programavimo kalbą *HTTP* serveriuose. Programavimo kalbos interpretatoriaus parsisiuntimas ir įdiegimas vyko sklandžiai.

Svetainės apdorojamų užklausų kiekis. Siekiant apskaičiuoti svetainės apdorojamų užklausų skaičių per vieną sekundę, yra naudojamas *Apache* įrankis *ab*. Šis įrankis yra naudojamas komanda `ab -n 100 -c 10 -k -H "Accept-Encoding: gzip, deflate`.

Kiekvienam nešiojam kompiuteriui yra atliekami 50 užklausų testų, kuriose yra vykdoma 100 užklausų, po 10 kiekvienu testu. Testų rezultatai yra sudedami ir padalijami iš įvykdytų testų skaičiaus. Testo rezultatai yra pateikiami 16 lentelėje.

16 lentelė. *Python* svetainės užklausų vidurkis vykdant 100 užklausų

Programavimo kalba	<i>Samsung NP350V5C</i> svetainės užklausų vidurkis per 1 sekundę	<i>Lenovo g510</i> svetainės užklausų vidurkis per 1 sekundę	<i>HP 15-n245sa</i> svetainės užklausų vidurkis per 1 sekundę
<i>Python</i>	7,99	8,02	13,04

Pagal pateiktą lentelę išsiaiškinome, kad *Samsung NP350V5C* nešiojamasis kompiuteris atliko 7,99 svetainės užklausų per sekundę, *Lenovo g510* sugebėjo atlikti 8,02 svetainės užklausas per sekundę, o *HP 15-n245sa* – svetainės 13,04 užklausas per sekundę.

Duomenų bazės poveikis svetainės naudotojų užklausų vykdymui. Tikrinamas *MySQL* duomenų bazės valdymo sistemos duomenų bazės *items* poveikis svetainės užklausų apdorojimo greičiui. Interneto svetainės kūrimo kalbos tikrinamos duomenų bazėje esant vienai eilutei, penkiasdešimt eilučių ir esant šimtui eilutėms. Testavimui naudojamas *Apache* įrankis *Apache Bench*. Testo rezultatai yra pateikiami 17 lentelėje.

17 lentelė. Duomenų bazės poveikis **Python** svetainės užklausų vykdymui

Duomenų bazės eilučių skaičius Python svetainėje	Samsung NP350V5C svetainės užklausų vidurkis per 1 sekundę	Lenovo g510 svetainės užklausų vidurkis per 1 sekundę	HP 15-n245sa svetainės užklausų vidurkis per 1 sekundę
1 eilutė	7,98	8,05	13,05
50 eilučių	7,95	8,01	12,97
100 eilučių	7,73	7,77	12,61

Lentelėje pavaizduoti kiekvieno nešiojamojo kompiuterio vidutinis svetainės užklausų apdorojimo kiekis per 1 sekundę, kai duomenų bazėje yra 1 eilutė, 50 eilučių ir 100 eilučių. Naudojantis šios lentelės duomenimis yra išvedama 18 lentelė. Maksimalus svetainės apdorojimo kiekis yra laikomas interneto svetainės kūrimo kalbos apdorojimo kiekis duomenų bazėje esant 1 įvesties eilutei.

18 lentelė. Procentinis duomenų bazės poveikis **Python** svetainės užklausų vykdymui

Duomenų bazės eilučių skaičius Python svetainėje	Samsung NP350V5C svetainės užklausų apdorojimo kiekis, %	Lenovo g510 svetainės užklausų apdorojimo kiekis, %	HP 15-n245sa svetainės užklausų apdorojimo kiekis, %
1 eilutė	100 %	100 %	100 %
50 eilučių	99,62 %	99,50 %	99,39 %
100 eilučių	96,87 %	96,52 %	96,63 %

Pakeitus duomenų bazės eilučių skaičių iki 50, nešiojamųjų kompiuterių svetainių apdorojimo efektyvumai sulėtėja. **Samsung** svetainės apdorojimo kiekis sumažėjo iki 99,62 %, **Lenovo** iki 99,50 %, o **Hp** – 99,39 %. Padidinus duomenų bazės eilučių skaičių iki 100, atitinkamai nešiojamųjų kompiuterių svetainių apdorojimo efektyvumas sumažėjo iki 96,87 %, 96,52 % ir 96,63 %.

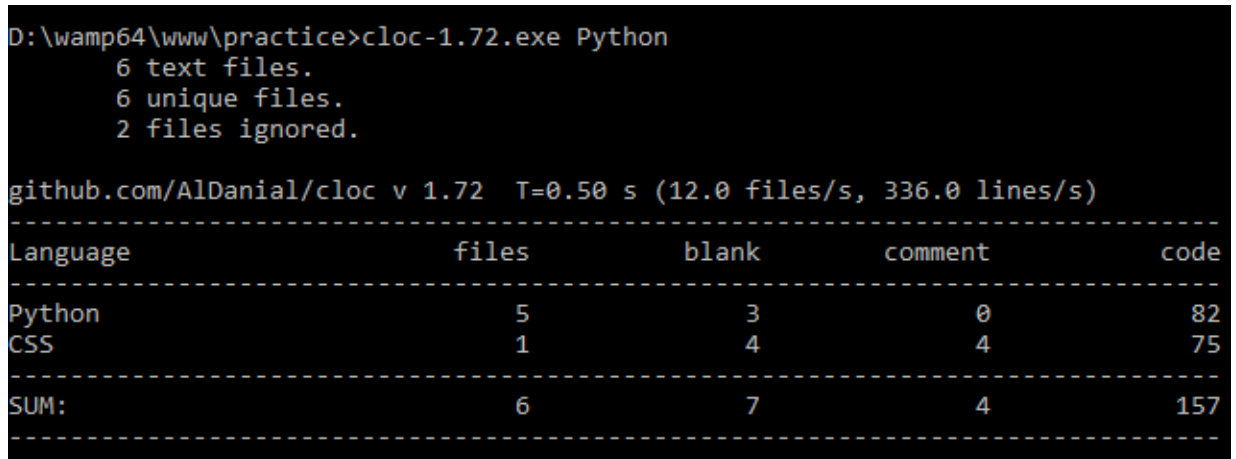
Svetainės apdorojamų užklausų poveikis procesoriui. Užklausų poveikis procesoriui yra tikrinamas **Apache** įrankio **ab** komanda `ab -n 100 -c 10 -k -H "Accept-Encoding: gzip, deflate"`, bei **Task Manager** programa. Yra atlieki 50 testų kiekvienam naudojimui nešiojamajam kompiuteriui. Testų rezultatai yra pateikiami 19 lentelėje.

19 lentelė. **Python** svetainės procesoriaus galios naudojimas vykdant 100 užklausų

Programavimo kalba	Samsung NP350V5C procesoriaus galios sunaudojimas, %	Lenovo g510 procesoriaus galios sunaudojimas, %	HP 15-n245sa procesoriaus galios sunaudojimas, %
Python	99,92 %	99,76 %	99,12 %

Tikrinant 100 **Python** interneto svetainės kūrimo kalbos užklausų išsiaiškinta, kad **Samsung NP350V5C** nešiojamasis kompiuteris vidutiniškai naudoja 99,92 % procesoriaus galios, **Lenovo g510** naudoja 99,76 % galios o **Hp 15-n245sa** nešiojamasis kompiuteris – 99,12 % procesoriaus galios.

Svetainės kodo eilučių skaičius. Naudojantis **Cloc** programine įranga yra skaičiuojamas **Python** svetainės kodo eilučių skaičius. Naudojama ta pati programos versija visuose programavimo kalbų bandymuose. **Cloc** aktyvuojama naudojantis **Command prompt** interpretatoriumi. Rezultatas yra pateikiamas 7 paveikslėlyje.



```
D:\wamp64\www\practice>cloc-1.72.exe Python
  6 text files.
  6 unique files.
  2 files ignored.

github.com/AlDanial/cloc v 1.72  T=0.50 s (12.0 files/s, 336.0 lines/s)
-----
Language             files      blank     comment       code
-----
Python                 5           3           0          82
CSS                     1           4           4          75
-----
SUM:                   6           7           4         157
-----
```

7 pav. **Python** programavimo kalbos svetainės kodo tipai ir eilučių skaičius

Norint apskaičiuoti tikslų svetainėje naudojamų kodo eilučių skaičių, iš bendro kodo eilučių skaičių, pateikto **SUM** eilutėje, yra atimamas bendras komentarų ir tuščių kodo eilučių skaičius. Išsiaiškiname, kad **Python** svetainės kūrimo kalbą sudaro 79 **Python** programavimo kalbos eilutės ir 67 **CSS** kodo eilutės. Programoje naudojamos **HTML** svetainių kūrimo kalba yra įskaičiuojama į **Python** eilučių skaičių. Galutinis svetainės kodą sudaro 146 **Python** ir **CSS** kodo eilutės.

4.1.3 Interneto svetainės kūrimo kalbos **Ruby** bandymai

Svetainės kalbos interpretatoriaus parsisiuntimas ir įdiegimas. Programavimo kalbos **Ruby** interpretatorius yra parsisiunčiamas iš oficialios programavimo kalbos svetainės <https://rubyinstaller.org>. Nurodytoje svetainėje yra pasirenkamas parsisiuntimo mygtukas, kuris peradresuoja naudotoją į kitą tinklapį, kuriama pasirenkama 2.3.3 64 bitų programavimo kalbos versija. Yra parsisiunčiamas **Ruby** programavimo kalbos paketas.

Aktyvavus programavimo kalbos paketą yra pradedamas interpretatoriaus instaliavimas. Instaliavimo metu yra pasirenkama norima įrašymo alokacija bei nurodoma programavimo kalbos interpretatoriaus paleidimo kelias. Siekiant naudoti **Ruby** interpretatorių interneto svetainių kūrimui yra naudojamas **Rack** serverio sąsaja. **Rack** yra įrašomas naudojantis **Command prompt** interpretatorių, naudojantis komanda `gem install rack`. Programavimo kalbos interpretatoriaus parsisiuntimas ir įdiegimas vyko sklandžiai.

Svetainės apdorojamų užklausų kiekis. Svetainės užklausų skaičius per sekundę yra apskaičiuojamas *Apache* įrankis *ab*. Naudojantis *Command prompt* kodo interpretatoriumi yra naudojama komanda `ab -n 100 -c 10 -k -H "Accept-Encoding: gzip, deflate`.

Kadangi kiekvienos užklausos rezultatai yra nepastovūs ir kintantys, yra vykdomi 50 testų kiekviename nešiojamame kompiuteryje. Kiekvieno testo metu yra vykdoma 100 svetainės užklausų, po 10 užklausų vienu metu. Išsiaiškinus kiekvieno prietaiso svetainės užklausų vidurkį per vieną sekundę, visų prietaisų užklausų vidurkis yra sudedamas ir padalijamas iš naudotų prietaisų skaičiaus. Testo rezultatai yra pateikiami 20 lentelėje.

20 lentelė. *Ruby* svetainės užklausų vidurkis vykdant 100 užklausų

Programavimo kalba	<i>Samsung NP350V5C</i> svetainės užklausų vidurkis per 1 sekundę	<i>Lenovo g510</i> svetainės užklausų vidurkis per 1 sekundę	<i>HP 15-n245sa</i> svetainės užklausų vidurkis per 1 sekundę
<i>Ruby</i>	49,52	50,58	80,99

Iš pateiktos lentelės matyti, kad daugiausia svetainės apklausų per vieną sekundę įvykdė *HP 15-n245sa* nešiojamasis kompiuteris, kurio vidutinis užklausų apdorojimo kiekis yra 80,99 užklausos per sekundę. *Samsung NP350V5C* vidutiniškai įvykdė 49,52 svetainės užklausas per vieną sekundę, o *Lenovo g510* – 50,58 svetainės užklausas per sekundę.

Duomenų bazės poveikis svetainės naudotojų užklausų vykdymui. Testuojant interneto svetainės kūrimo kalbos svetainę yra tikrina duomenų bazės *MySQL* poveikis svetainės apdorojimo greičiui. Testuojant duomenų bazės poveikį svetainei, duomenų bazės eilučių skaičius yra keičiamas. Testavimas vyksta kai duomenų bazėje yra 1 eilutė, 50 eilučių ir 100 eilučių. Testavimui naudojamas *Apache* įrankis *Apache Bench*. Testo rezultatai yra pateikiami 21 lentelėje.

21 lentelė. Duomenų bazės poveikis *Ruby* svetainės užklausų vykdymui

Duomenų bazės eilučių skaičius <i>Ruby</i> svetainėje	<i>Samsung NP350V5C</i> svetainės užklausų vidurkis per 1 sekundę	<i>Lenovo g510</i> svetainės užklausų vidurkis per 1 sekundę	<i>HP 15-n245sa</i> svetainės užklausų vidurkis per 1 sekundę
1 eilutė	7,98	8,05	13,05
50 eilučių	7,95	8,01	12,97
100 eilučių	7,73	7,77	12,61

Naudojantis šios lentelės duomenimis yra išvedama 22 lentelė. Maksimalus svetainės apdorojimo kiekis yra laikomas interneto svetainės kūrimo kalbos apdorojimo kiekis duomenų bazėje esant 1 įvesties eilutei.

22 lentelė. Procentinis duomenų bazės poveikis **Ruby** svetainės užklausų vykdymui

Duomenų bazės eilučių skaičius Ruby svetainėje	Samsung NP350V5C svetainės užklausų apdorojimo kiekis, %	Lenovo g510 svetainės užklausų apdorojimo kiekis, %	HP 15-n245sa svetainės užklausų apdorojimo kiekis, %
1 eilutė	100 %	100 %	100 %
50 eilučių	96,07 %	95,21 %	96,78 %
100 eilučių	86,158 %	87,14 %	89,53 %

Pakeitus duomenų bazės eilučių skaičių iki 50, nešiojamųjų kompiuterių svetainės apdorojimo efektyvumai sulėtėja. **Samsung** svetainės apdorojimo kiekis nukrito iki 96,07 %, **Lenovo** sumažėjo iki 95,21 %, o **Hp** – 96,78 %. Padidinus duomenų bazės eilučių skaičių iki 100, atitinkamai nešiojamųjų kompiuterių svetainės apdorojimo efektyvumas nukrito iki 86,58 %, 87,14 % ir 89,53 %.

Svetainės apdorojamų užklausų poveikis procesoriui. Svetainės apdorojamų užklausų poveikis procesoriui yra tikrinamas atliekant 100 interneto svetainės kūrimo kalbos svetainės užklausų **Command prompt** interpretatoriuje, naudojantis **Apache** įrankiu **ab**. Informacija yra fiksuojama naudojantis **Task Manager** programa. Kiekvienam nešiojamajam kompiuteriui yra atliekami 50 bandymų. Testų rezultatai yra pateikiami 23 lentelėje.

23 lentelė. **Ruby** svetainės procesoriaus galios naudojimas vykdant 100 užklausų

Programavimo kalba	Samsung NP350V5C procesoriaus galios sunaudojimas, %	Lenovo g510 procesoriaus galios sunaudojimas, %	HP 15-n245sa procesoriaus galios sunaudojimas, %
Ruby	38,42 %	35,28 %	31,64 %

Išsiaiškinta, kad daugiausiai galios sunaudoja **Samsung NP350V5C** nešiojamasis kompiuteris, kuris sunaudoja 38,42 %. **Lenovo g510** sunaudoja 35,28 % procesoriaus galios, o **Hp 15-n245sa** nešiojamasis kompiuteris sunaudoja mažiausiai – 31,64 % procesoriaus galios. Procesorių galios sunaudojimų vidurkis yra 35,11 %.

Svetainės kodo eilučių skaičius. **Ruby** svetainės kodo eilučių skaičius yra tikrinamas naudojantis programa **Cloc**. Ši programa yra parsisiunčiama iš <https://github.com/AIDanial/cloc>. Programa aktyvuojama naudojantis **Command prompt** interpretatoriumi. Rezultatas yra pateikiamas 8 paveikslėlyje.

```
D:\wamp64\www\practice>cloc-1.72.exe Ruby
8 text files.
8 unique files.
6 files ignored.
```

github.com/AlDanial/cloc v 1.72 T=1.00 s (7.0 files/s, 212.0 lines/s)

Language	files	blank	comment	code
Ruby	5	7	0	86
CSS	1	4	4	75
ERB	1	0	0	36
SUM:	7	11	4	197

8 pav. *Ruby* programavimo kalbos svetainės kodo tipai ir eilučių skaičius

Kuriant *Ruby* kalbos programą iškilo problemų naudojant *HTML* programavimo kalbą. Siekiant išs্পesti šią problemą buvo sukurta šablonas, kuriame pateikiamas *HTML* kodas su *ERB* kodo elementais. *ERB* yra priskaičiuojamas prie bendro kodo eilučių skaičiaus. Dėl naudojamo šablono programos failų skaičius padidėjo vienu skaitmeniu.

Bendras *Ruby* programoje panaudoto kodo eilučių skaičius yra apskaičiuojamas iš bendro kodo eilučių kiekio atimant komentarų ir tuščių kodo eilučių skaičių. Į bendrą kodo eilučių skaičių yra įskaičiuotas *Ruby*, *CSS* ir *ERM* kodo eilutės. Svetainės kode yra 79 *Ruby* kodo eilutės, 68 *CSS* kodo eilutės ir 36 *ERB* kodo eilutės. *Ruby* programavimo kalba naudoja 182 kodo eilutes.

4.2. Interneto svetainių kūrimo kalbų palyginamosios analizės bandymų rezultatai

Ankstesniuose poskyriuose buvo atlikti interneto svetainių kūrimo kalbų *PHP*, *Python* ir *Ruby* bandymai. Šiame skyriuje aptariami ir apibendrinami gauti bandymų rezultatai. Bandymų rezultatai yra pateikiami 24 lentelėje.

24 lentelė. Interneto svetainių kūrimo kalbų bandymų rezultatai

Nr.	Kriterijus	PHP	Python	Ruby
1	Svetainės kalbos interpretatoriaus parsisiuntimas ir įdiegimas	2	2	2
2	Svetainės Apdorojamų užklausų kiekis	3	1	2
3	Duomenų bazės poveikis svetainių naudotojų užklausų vykdymui	1	3	2
4	Svetainių apdorojamų užklausų poveikis procesoriams	3	1	2
5	Svetainės kodo eilučių skaičius	2	3	1
	<i>Samsung NP350V5C</i>	155	146	182
	Iš viso:	11	10	9

Sudarant bendrą interneto svetainių kūrimo kalbų **PHP**, **Python** ir **Ruby** bandymo rezultatų lentelę buvo naudojama informacija iš 1 – 23 lentelių bei 6 – 8 paveikslėlių.

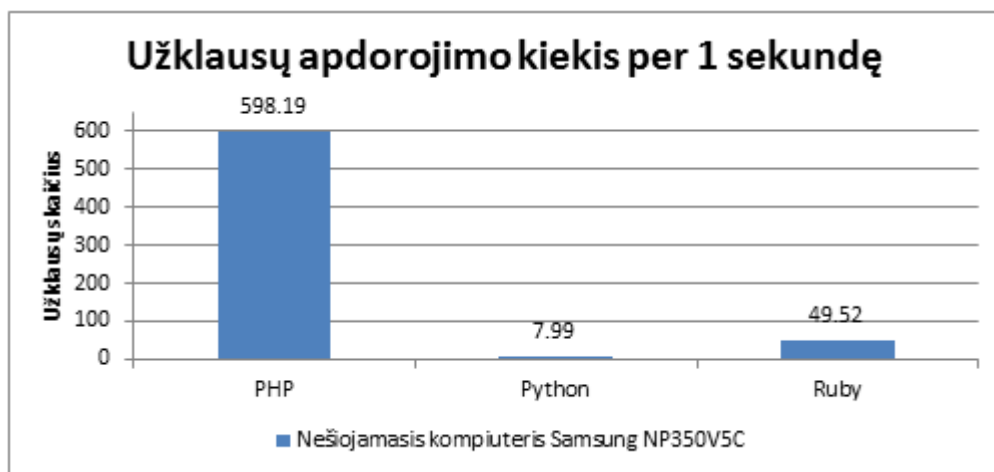
Parsiuntimas ir įdiegimas. 1 – 3 bandymas. Interneto svetainės kūrimo kalbos **PHP** interpretatorius parsiumčiamas iš <http://www.wampserver.com>. Programavimo kalbos interpretatorius yra įrašoma kartu su **Wamp** serveriu, **Apache** programinė įranga ir **MySQL** duomenų bazė. Įrašoma **PHP** 5.6.25 versija. Įdiegimo pabaigoje pasirenkama žiniatinklio naršyklė **Chrome**. Programavimo kalbos interpretatoriaus parsiumimas ir įdiegimas vyko sklandžiai. Skiriami 2 balai.

4 – 6 bandymas. Interneto svetainės kūrimo kalbos **Python** interpretatorius yra parsiumčiamas iš oficialaus interneto svetainės <https://www.python.org>. Svetainėje pasirenkama 3.5.0 interneto svetainės kūrimo kalbos interpretatorius versija. Parsiumtus programavimo kalbos interpretatoriaus yra vykdomas kalbos įrašymas. Pasibaigus įrašymo procesui yra koreguojamas **Apache** `httpd.conf` failas. Programavimo kalbos interpretatoriaus parsiumimas ir įdiegimas vyko sklandžiai. Skiriami 2 balai.

7 – 9 bandymas. Programavimo kalbos **Ruby** interpretatorius yra parsiumčiamas iš oficialios programavimo kalbos svetainės <https://rubyinstaller.org>. Parsiumčiama 2.3.3 64 bitų programavimo kalbos interpretatoriaus versija.

Instaliavimo metu yra pasirenkama norima įrašymo alokacija bei nurodoma programavimo kalbos interpretatoriaus paleidimo kelias. Su programavimo kalbos interpretatoriumi yra naudojamas **Rack** serverio sąsaja. **Rack** yra įrašomas naudojantis **Command prompt** interpretatoriumi, komanda `gem install rack`. Programavimo kalbos interpretatoriaus parsiumimas ir įdiegimas vyko sklandžiai. Skiriami 2 balai.

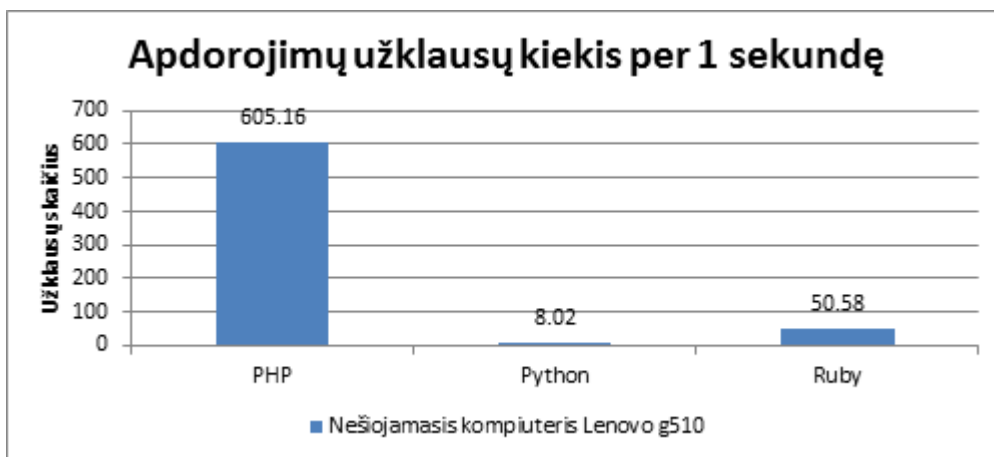
Svetainės apdorojamų užklausų kiekis. 10, 13, 16 bandymai. Tikrinamas nešiojamojo kompiuterio **Samsung NP350V5C** programavimo kalbų svetainių apdorojimų užklausų kiekį per vieną sekundę. Bandymų rezultatai yra pateikiami 9 paveikslėlyje.



9 pav. Užklausų apdorojimo kiekis per 1 sekundę kompiuteryje *Samsung NP350V5C*

Tikrinant svetainių apdorojamų užklausų kiekį **PHP** interneto svetainės kūrimo kalba apdoroja daugiausia užklausų – vidutiniškai 598,19 užklausas per sekundę. Antroje vietoje su 49,52 svetainės užklausomis per sekundę yra **Ruby** programavimo kalba. Mažiausiai svetainės užklausų per sekundę apdoroja **Python** programavimo kalba su 7,99 užklausomis per sekundę. Tikrinant svetainės apdorojamų užklausų kiekį su problemomis nesusidurta, bandymai įvyko sėkmingai.

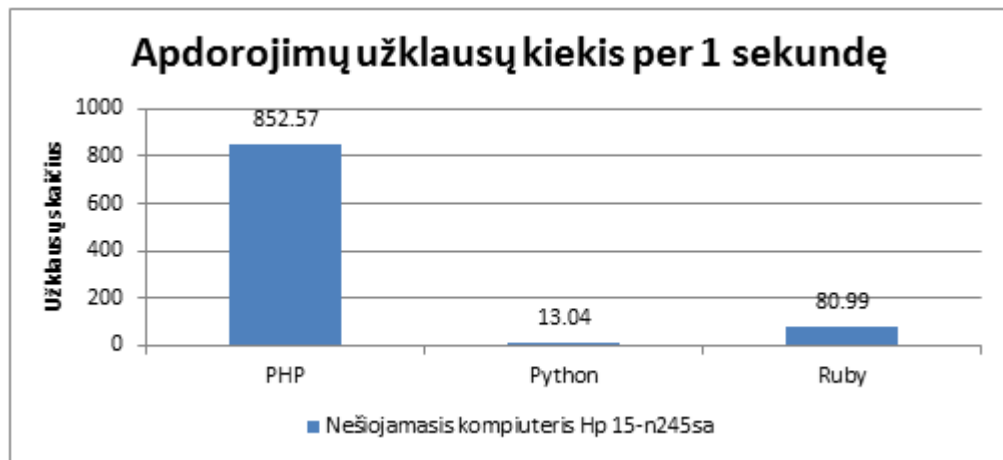
11, 14, 17 bandymai. Tikrinamas nešiojamojo kompiuterio **Lenovo g510** svetainių apdorojimų užklausų kiekį per vieną sekundę. Bandymų rezultatai yra pateikiami 10 paveikslėlyje.



10 pav. Užklausų apdorojimo kiekis per 1 sekundę kompiuteryje *Lenovo g510*

Naudojantis **Lenovo g510** nešiojamuoju kompiuteriu **PHP** programavimo kalbos svetainė apdoroja vidutiniškai 605,16 svetainės naudotojų užklausų per sekundę. **Python** programavimo kalba apdoroja vidutiniškai 8,02 svetainės apklausų per sekundę, o **Ruby** – 50,58 svetainės užklausų per sekundę. Vykdamas apdorojimo užklausus su problemomis nesusidurta, bandymai įvyko sėkmingai.

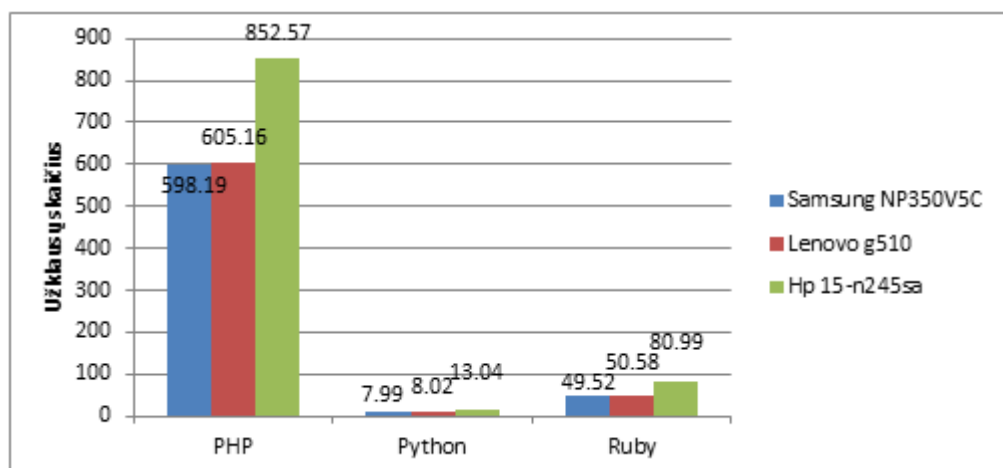
12, 15, 18 bandymai. Svetainės užklausų apdorojimo kiekio per sekundę rezultatai yra pateikiami 11 paveikslėlyje.



11 pav. Užklausų apdorojimo kiekis per 1 sekundę kompiuteryje *Hp 15-n245sa*

Su nešiojamuoju kompiuteriu *Hp 15-n245sa* programavimo kalba **PHP** apdoroja daugiausia svetainių užklausų per sekundę – vidutiniškai 852,57 užklausa per sekundę. Antroje vietoje yra **Ruby** programavimo kalba, kuri apdoroja vidutiniškai 80,99 svetainės užklausa per sekundę. Paskutinėje vietoje yra **Python** kalba, kuri apdoroja vidutiniškai 13,04 svetainės užklausa per sekundę. Vykdamas apdorojimo užklausa su problemomis nesusidurta, bandymai įvyko sėkmingai.

Apdorojamų užklausų kiekio rezultatai yra pateikiami 12 paveikslėlyje.



12 pav. Apdorojamų užklausų kiekio rezultatai

Analizuojant apdorojamų užklausų kiekio rezultatą matome kad **PHP** programavimo kalba apdoroja didžiausią skaičių svetainės naudotojų užklausų per sekundę. Antroje vietoje pagal apdorojamą užklausų kiekį yra **Ruby** interneto svetainės kūrimo kalba, trečioje – **Python**.

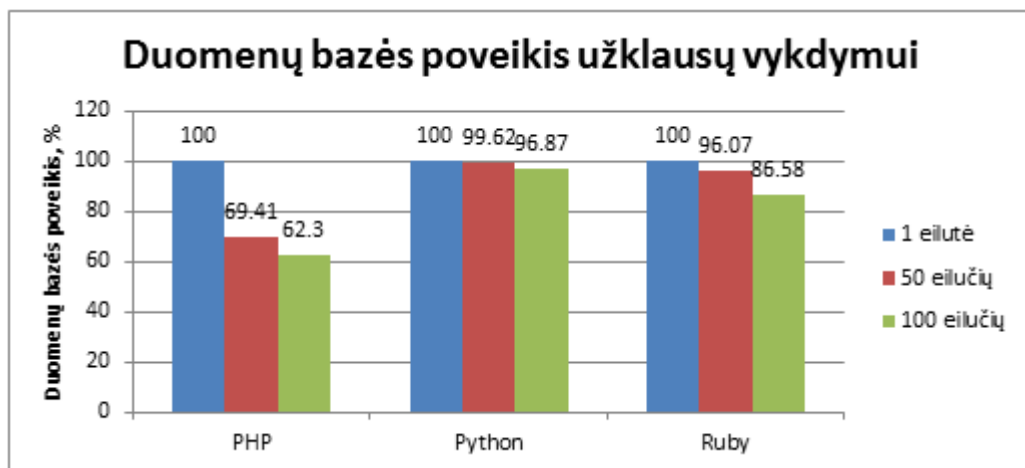
PHP pranašumo priežastis yra tai, kad **WAMP** serveris yra skirtas **PHP** interneto svetainėms. Nors **Python** programavimo kalbą galima naudoti **WAMP** serveryje, šios kalbos svetainės užklausa

greitis neprilygsta **PHP** kalbai. Taip pat **Python** naudoja `py.cgi` modulį, kuris yra skirtas lėtesnėms interneto svetainėms.

Kitaip nei **PHP** ir **Python**, **Ruby** naudoja **Rack** serverio sąsaja, kuri įtakoja **Ruby** svetainės apdorojimo užklausų kiekį. Dėl šios priežasties **Ruby** programavimo kalbos apdorojimo greitis yra aukštesnis už **Python**.

Išanalizavus skirtingų kompiuterių rezultatus pastebėta, kad **Hp 15-n245sa** nešiojamasis kompiuteris svetainės apdorojimo greitis yra aukštesnis už **Samsung NP350V5C** ir **Lenovo g510** nešiojamuosius kompiuterius. Taip yra todėl, kad **Hp** nešiojamojo kompiuterio procesorius ir operatyvioji atmintis yra geresnė nei kitų nešiojamųjų kompiuterių.

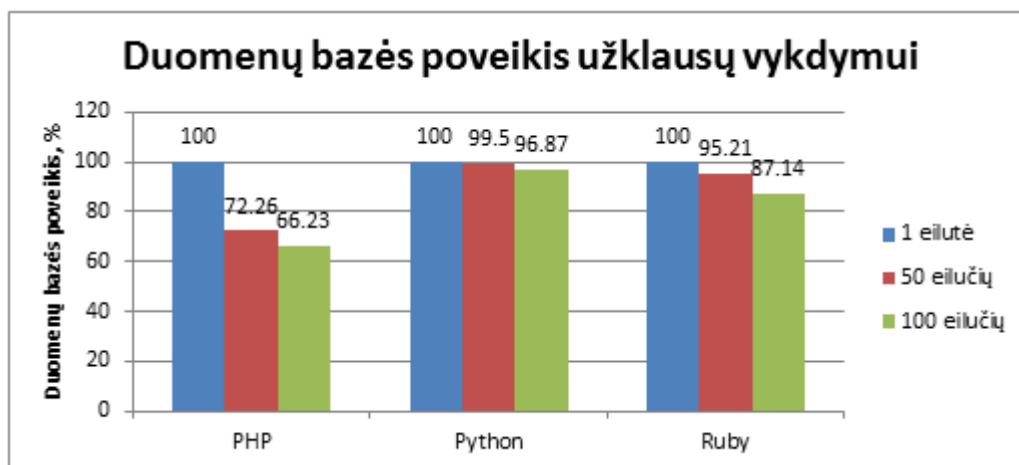
Duomenų bazės poveikis svetainės naudotojų užklausų vykdymui. 19, 22, 25 bandymai. Tikrinamas **Samsung NP350V5C** nešiojamojo kompiuterio `items` duomenų bazės dydžio poveikis užklausų apdorojimo kiekiui. Duomenų bazės poveikio svetainės užklausų vykdymui rezultatai yra pateikiami 13 paveikslėlyje.



13 pav. Duomenų bazės poveikis užklausų vykdymui kompiuteryje **Samsung NP350V5C**

Interneto svetainė testuojama trejose stadijose: duomenų bazėje esant vienai eilutei, penkiasdešimčiai eilučių ir šimtui eilučių. **PHP** programavimo kalbos svetainės užklausų apdorojimo efektyvumas sumažėja 31,59 % vykdant svetainę su duomenų baze, kurioje yra 50 eilučių duomenų. Padidinus eilučių skaičių iki 100, kalbos efektyvumas krenta 37,70 %. **Python** programavimo kalbos efektyvumo pakitimas yra nežymus, padidinus duomenų bazės eilučių skaičių iki 50 ir 100 eilučių, svetainės efektyvumas sumažėja atitinkamai 0,38 % ir 3,13 %. **Ruby** programavimo kalbos svetainės efektyvumas duomenų bazėje esant 50 eilučių sumažėja 3,93 %, 100 eilučių – 13,42 %.

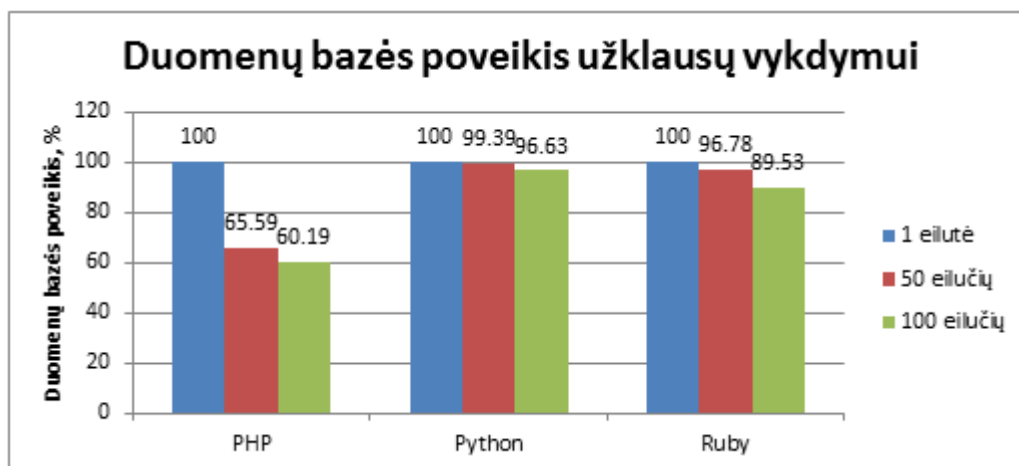
20, 23, 26 bandymai. Duomenų bazės poveikio svetainės užklausų vykdymui rezultatai yra pateikiami 14 paveikslėlyje.



14 pav. Duomenų bazės poveikis užklausų vykdymui kompiuteryje *Lenovo g510*

Lenovo g510 nešiojamojo kompiuterio *MySQL* duomenų bazės dydžio poveikis *PHP* kalbai esant 50 duomenų eilutėms efektyvumas sumažėja 27,74 %. Esant šimtui eilučių – 33,77 %. *Python* svetainės efektyvumas sumažėja 0,50 % padidinus duomenų bazę iki 50 duomenų eilučių. Esant šimtui eilučių kalbos svetainė sulėtėja 3,13 %. *Ruby* programavimo kalba sulėtėja 4,79 % ir 12,86 % esant duomenų bazėje 50 ir 100 eilučių.

21, 24, 27 bandymai. Tikrinamas *Hp 15-n245sa* nešiojamojo kompiuterio *MySQL* duomenų bazės dydžio poveikį svetainės užklausų apdorojimo kiekiui. Bandymo rezultatai yra pateikiami 15 paveikslėlyje.



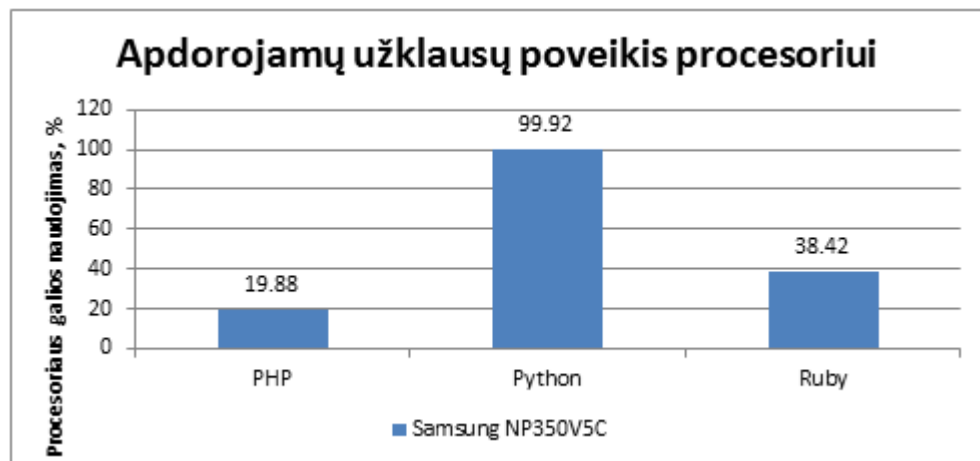
15 pav. Duomenų bazės poveikis užklausų vykdymui kompiuteryje *Hp 15-n245sa*

Interneto svetainės kūrimo kalbos *PHP* svetainės užklausų apdorojimo efektyvumas sumažėja 34,41 % duomenų bazėje esant 50 duomenų eilučių. Esant šimtui eilučių, efektyvumas nukrinta 39,81 %. Duomenų bazėje esant 50 ir 100 eilučių, *Python* programavimo kalbos darbingumas atitinkamai nukrinta 0,61 % ir 3,37 %. *Ruby* programavimo kalbos darbingumas nukrinta 3,22 %

duomenų bazėje esant 50 eilučių duomenų. Esant 100, **Ruby** kalbos efektyvumas sumažėja 10,47 %.

Naudojantis duomenų bazės poveikio užklausų vykdymo rezultatais skirtingose nešiojamuosiuose kompiuteriuose pastebėta, kad **Python** programavimo kalbos efektyvumas nukrinta mažiausiai, sekančioje vietoje pagal kalbos efektyvumą yra **Ruby** svetainės kūrimo kalba, trečioje – **PHP** svetainės kūrimo kalba.

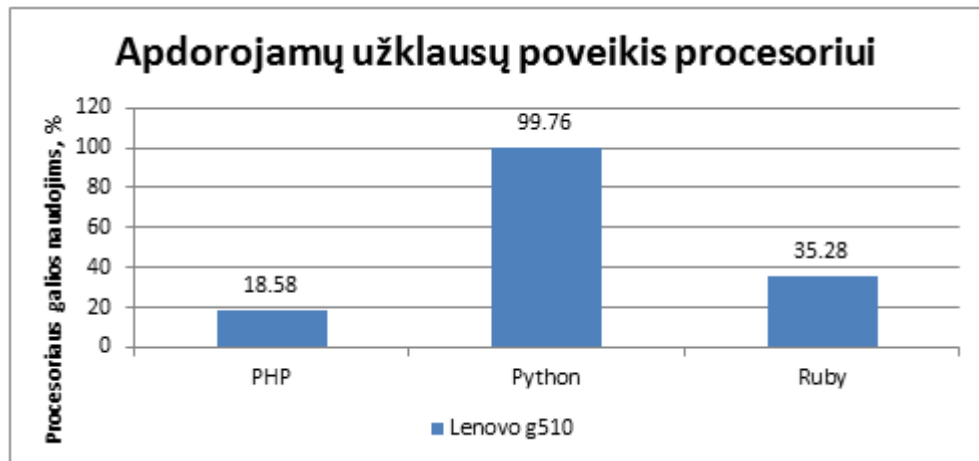
Svetainės apdorojamų užklausų poveikis procesoriui. 28, 31, 34 bandymai. Tikrinamas svetainės apdorojamų užklausų poveikis **Samsung NP350V5C** nešiojamojo kompiuterio procesoriui. Šiam bandymui įvykdyti yra naudojamas **Apache** įrankis **ab** bei **Task Manager**. Apdorojamų svetainės užklausų poveikis procesoriui rezultatas yra pateikiamas 16 paveikslėlyje.



16 pav. Apdorojamų užklausų poveikis kompiuterio **Samsung NP350V5C** procesoriui

Testuojant svetainės apdorojamų užklausų poveikį **Samsung NP350V5C** kompiuterio procesoriui pastebėta, kad programavimo kalbos **PHP** svetainė vidutiniškai naudoja 19,88 % procesoriaus galios. **Ruby** vidutiniškai naudoja 38,42 % procesoriaus galios, o **Python** – net 99,92 % procesoriaus galios.

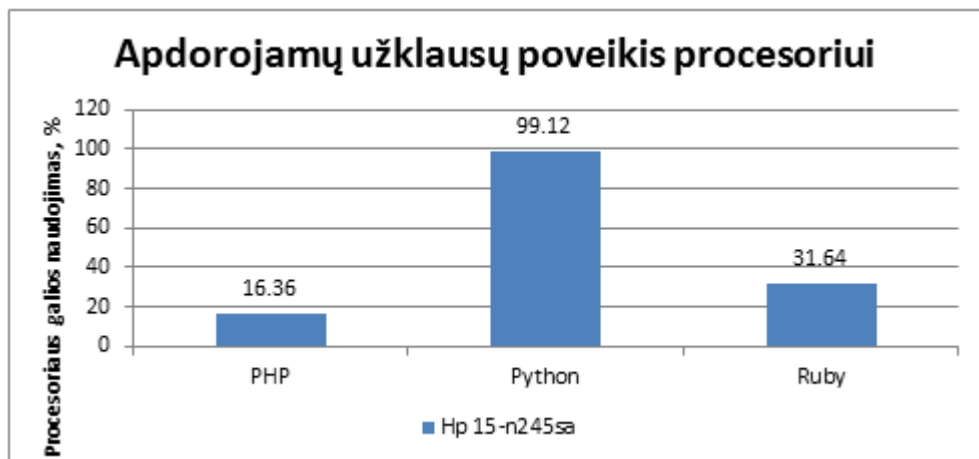
29, 32, 35 bandymai. Apdorojamų užklausų poveikis kompiuterio **Samsung NP350V5C** procesoriui rezultatas yra pateikiamas 17 paveikslėlyje.



17 pav. Apdorojamų užklausų poveikis kompiuterio *Lenovo g510* procesoriui

PHP svetainė vykdanči svetainės naudotojų užklausas su *Lenovo g510* kompiuteriu pastebėta, kad ši kalba vidutiniškai naudoja 18,58 % procesoriaus galios. **Ruby** programavimo kalba vidutiniškai naudoja 35,28 %, o **Python** – 99,76 %.

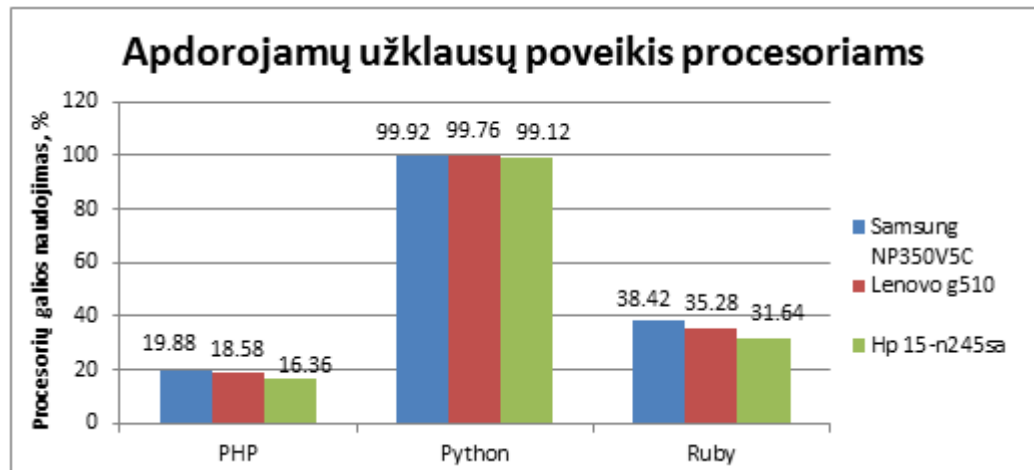
30, 33, 36 bandymai. Testuojamas svetainės apdorojamų užklausų poveikis *Hp 15-n245sa* nešiojamojo kompiuterio procesoriui. Bandymo rezultatai yra pateikiami 18 paveikslėlyje.



18 pav. Apdorojamų užklausų poveikis kompiuterio *Hp 15-n245sa* procesoriui

Bandymu metu pastebėta, kad **PHP** svetainė naudoja vidutiniškai 16,36 % procesoriaus galios. Daugiausiai procesoriaus galios naudoja **Python** programavimo kalba – net 99,12 % galios. **Ruby** vidutiniškai naudoja 31,64 % nešiojamojo kompiuterio procesoriaus galios.

Svetainės apdorojamų užklausų poveikiai procesoriams bandymų rezultatai yra pateikiami 19 paveikslėlyje.



19 pav. Apdorojamų užklausų poveikis procesoriams rezultatai

Testuojant interneto svetainių kūrimo kalbų svetaines nešiojamaisiais kompiuteriais **Samsung NP350V5C**, **Lenovo g510** ir **Hp 15-n245sa** sužinota, kad **PHP** programavimo kalba naudoja mažiausiai procesoriaus galios apdorojant naudotojų užklausas. Antroje vietoje pagal galios sunaudojimą yra **Ruby** programavimo kalba. Daugiausiai procesorių galios sunaudoja **Python** programos svetainė.

Patikrinus interneto svetainių kūrimo kalbų interpretatorių failus buvo sužinota, kad **Python** naudoja `mod_cgi` modulį. Šis modelis yra skirtas naudoti svetainėse su mažesniu užklausų srautu. Dėl didelio skaičiaus naudotojų užklausų naudojama visa procesoriaus galia.

Svetainės kodo eilučių skaičius. 37 – 39 bandymai. Iš 37 – 39 bandymo matyti, kad mažiausią svetainės kodo eilučių skaičių turi **Python** programavimo kalba – programavimo kalbos svetainė sudaryta iš 146 kodo eilučių. Antroje vietoje pagal svetainės kodo eilučių skaičių yra **PHP** interneto svetainės kalba. **PHP** interneto svetainių kūrimo kalbos svetainė sudaro 155 kodo eilutės. **Ruby** programavimo kalbos svetainę sudaro 182 kodo eilutės.

4.3. Skyriaus apibendrinimas

Aprašius interneto svetainių kūrimo kalbų palyginamosios analizės eigą, apibendrinus ir aptarus rezultatus sužinota, kad interneto svetainių kūrimo kalbos atliko visas skirtas užduotis.

Parsiunčiant ir įdiegiant interneto svetainių kūrimo kalbas **PHP**, **Ruby** ir **Python** nebuvo susidurta su problemomis.

Apdorojamų užklausų kiekis per sekundę buvo matuojamas atliekant 50 bandymų **PHP**, **Python** ir **Ruby** programavimo kalbai kiekviename kompiuteryje. Pastebėta, kad **PHP** programavimo kalba apdoroja daugiausiai užklausų per sekundę, sekančioje vietoje yra **Ruby**, trečioje – **Python**.

Testuojant duomenų bazės poveikį naudotojų užklausų vykdymui buvo atlikta 60 bandymų kiekvienai programavimo kalbai ant **Samsung NP350V5C**, **Lenovo g510** ir **Hp 15-n245sa** nešiojamųjų kompiuterių. Bandymai buvo suskirstyti į 3 stadijas: atlikta 20 bandymų duomenų bazėje esant 1 duomenų eilutei, 50 duomenų eilučių ir 100 duomenų eilučių. Pastebėta, kad **Python** programavimo kalbos užklausų apdorojimo greitis pakito minimaliai, pakeitus duomenų bazės dydį nuo 1 eilutės iki 100 eilučių užklausų vykdymo efektyvumas sumažėja mažiau nei 5 procentais. Antroje vietoje pagal pakitusi užklausų efektyvumą yra **Ruby** programavimo kalba. Didžiausias užklausų efektyvumo sumažėjimas pastebėtas **PHP** programavimo kalboje.

Matuojant užklausų poveikį nešiojamųjų kompiuterių procesoriams buvo atlikti 50 **PHP**, **Python** ir **Ruby** bandymų skirtinguose nešiojamuosiuose kompiuteriuose. Bandymu metu sužinota, kad daugiausiai procesorių galios sunaudoja **Python** programavimo kalbos svetainė, mažiausiai – **PHP** programavimo kalbos svetainė. Antroje vietoje pagal procesorių galios sunaudojimą yra **Ruby** programavimo kalbos svetainė. Pastebėta, kad **Python** naudoja maksimalią procesorių galią apdorojant naudotojų užklausas.

Tikrinant interneto svetainių kūrimo kalbų kodo eilučių skaičių sužinota, kad mažiausiai kodo eilučių naudoja **Python** programavimo kalbos svetainė. Antroje vietoje pagal kodo eilučių skaičių yra **PHP** programavimo kalbos svetainė. Daugiausiai kodo eilučių naudoja **Ruby** programavimo kalbos svetainė.

Remiantis 24 lentelės duomenimis nustatyta, kad pirma vieta atiteko interneto svetainių kūrimo kalbai **PHP**, kuri surinko 11 balų. Antra vieta skiriama **Python** programavimo kalbai – 10 balų. Trečioje vietoje liko **Ruby** programavimo kalba surinkusi 9 balus.

5. REKOMENDACIJOS DĖL INTERNETO SVETAINIŲ KŪRIMO KALBŲ NAUDOJIMO IŠVADOS IR PASIŪLYMAI

Atlikus **PHP**, **Python** ir **Ruby** palyginamąją analizę nustatyta, kad šios kalbos pasižymi skirtingomis funkcijomis. Programavimo kalba **PHP** išsiskiria vidutiniu naudotojų užklausų kiekiu per sekundę. Iš 10 – 18 bandymų rezultatų matyti, kad **PHP** interneto svetainių kūrimo kalbos svetainė veikdama ant skirtingų nešiojamųjų kompiuterių **Samsung NP350V5C**, **Lenovo g510** ir **Hp 15-n245sa** apdorojimo kiekis vidutiniškai kinta nuo 598,19 iki 852,57. Kuriant interneto svetainę kurioje tikimasi didelis naudotojų užklausų kiekis **PHP** interneto svetainių kūrimo kalba yra tinkamas programavimo kalbos variantas.

Testuojant apdorojamų užklausų kiekį, duomenų bazės poveikį užklausų vykdymui bei užklausų poveikį procesoriams sužinota, kad nors **Ruby** programavimo kalba neišsiskyrė išvardytuose bandymų laukuose, ši programavimo kalba neatskleidė akivaizdžių silpnųjų. Vykdamas apdorojimo užklausų kiekio ir užklausų poveikį procesoriams **Ruby** programavimo kalbos rezultatai aplenkė **Python** programavimo kalbą, testuojant duomenų bazės poveikį užklausų vykdymui – **PHP** programavimo kalbą. Iš rezultatų matyti kad nors **Ruby** nepirmauja nė vienoje iš išvardytų kriterijų ši programavimo kalba yra tinkama naudojimui svetainėse, kurios yra sukurtos atlikti skirtingus procesus.

Interneto svetainių kūrimo kalba **Python** pasižymi efektyvumu apdorojant dideles apimties duomenų bazių ryšius. 19 – 27 bandymo metu pastebėta, kad **Python** programavimo kalbos efektyvumas nukrinta mažiau nei 4 procentais pakitus duomenų bazės dydį nuo 1 duomenų eilutės iki 100 duomenų eilučių. Testuojant kitas programavimo kalbas pastebėta, kad **Ruby** programavimo kalbos efektyvumas sumažėja nuo 11 % iki 14 %, o **PHP** 34 – 40 %. Svetainės, kurios naudoja dideles duomenų bazes rekomenduojama naudoti **Python** interneto svetainių kūrimo kalbą.

Norint padidinti **Python** interneto svetainės apdorojamų apklausų kiekį bei sumažinti kompiuterių procesorių galios sąnaudas reikalinga pakeisti **Python** programavimo kalbos interpretatoriaus `mod_cgi` modulį į `mod_wsgi` modulį. `mod_cgi` modulis yra naudojamas svetainėse su mažesniu užklausų srautu.

IŠVADOS

1. Aptariamos internetas, interneto svetainės ir kūrimo kalbos sampratų reikšmės. Pagal prieš tai išvardytas sampratas aptariama, kad interneto svetainių kūrimo kalba yra programavimo kalba, naudojama kuriant internetinių tinklalapių rinkinį pasaulinėje kompiuterių tinklų sistemoje.

2. Apžvelgta interneto svetainės struktūra, aptarti du svetainės kūrimo būdai: svetainės kūrimas naudojantis **HTML** programavimo kalba ir turinio valdymo sistema. Naudojantis kriterijų lentelė sužinota, kad svetainės kūrimas **HTML** pagrindu leidžia didesnę modifikavimo galimybę, tačiau lengvesnė ir greitesnė yra turinio valdymo sistema.

3. Parengta interneto svetainių kūrimo kalbų **PHP**, **Python** ir **Ruby** palyginamosios analizės atlikimo metodika, nustatyta interneto svetainių kūrimo kalbų kriterijai. Nuspręsta, kad naudojami kriterijai yra lygiaverčiai ir kiekvienas iš jų sudaro 20% kriterijų vertinimo balų. Aptarta bandymuose naudojama įranga. Aprašoma sudaryta interneto svetainių kūrimo kalbų bandymų atlikimo procedūros matrica.

4. Aprašyta interneto svetainių kalbų **PHP**, **Python** ir **Ruby** palyginamosios analizės atlikimo eiga, apibendrinti ir aptarti rezultatai. Nustatyta, kad visų programavimo kalbų parsisiuntimas ir įdiegimas įvyko be nesklandumų. **PHP** interneto svetainės kūrimo kalba apdoroja daugiausia naudotojų užklausų per sekundę, **Python** programavimo kalba pasižymi efektyvumu apdorojant dideles duomenų bazes. **Ruby**, nors nepasižymi išskirtine savybe, neturi išskirtinių silpnųjų. Pastebėta, kad **Python** programavimo kalba naudoja maksimalią procesorių apdorojimo galią. Tikrinant programavimo kalbų eilučių skaičių sužinota, kad **Python** programavimo kalbos svetainė yra sudaryta iš mažiausiai kodo eilučių. Pagal parengtą palyginamosios analizės metodikos iškeltus kriterijus, nustatyta kad interneto svetainių kūrimo kalba **PHP** surinko 11 balų. Programavimo kalba **Python** surinko 10 balų, **Ruby** – 9 balus.

5. Pateiktos **PHP**, **Python** ir **Ruby** kalbų rekomendacijos dėl svetainių kūrimo kalbų naudojimo. Nustatyta, kad **PHP** interneto svetainės kūrimo kalba tinkama naudoti svetainėse su dideliu kiekiu naudotojų užklausų, **Python** – su didelėmis duomenų bazėmis. **Ruby** nepasižymi stiprybėmis bet neturi ir silpnųjų. Ši programavimo kalba tinkama naudoti svetainėse, kurios atlieka skirtingus procesus. Norint padidinti **Python** programavimo kalbos užklausų apdorojimo kiekį bei sumažinti naudojamų kompiuterių procesorių galios sąnaudas reikalinga pakeisti programavimo kalbos interpretatoriaus modulį `mod_cgi` į modulį `mod_wsgi`.

INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. Achour, M., Betz, F., Dovgal, A. *PHP manual* [interaktyvus]. PHP documentation group. 2017 [Žiūrėta 2017 m. Kovo 7 d.]. Prieiga per internetą: <http://php.net/manual/en>
2. Cert litnet. Pagrindiniai web sistemų pažeidžiamumai ir saugos būdai [interaktyvus]. 2015 [Žiūrėta 2017 m. Gegužės 23 d.]. Prieiga per internetą:
3. Cleverism. *Top programing languages used in web development* [interaktyvus]. 2015 [Žiūrėta 2016 m. Birželio 12 d.]. Prieiga per internetą: <https://www.cleverism.com/programming-languages-web-development/>
4. Dagienė, V., Grigas, G., Jevsikova, T. *Enciklopedinis kompiuterijos žodynas*. Vilnius: Matematikos ir informatikos institutas, 2008.
5. Danial, A. *Cloc* [interaktyvus]. 2015 [Žiūrėta 2017 m. Gegužės 3 d.]. Prieiga per internetą: <https://github.com/AIDanial/cloc>
6. Fedora Magazine. *Help port Python packages to Python 3* [interaktyvus]. 2015 [Žiūrėta 2017 m. Kovo 23 d.]. Prieiga per internetą: <https://fedoramagazine.org/help-port-python-packages-python-3>
7. Gližeris, G. Biuro administravimas. *Turinio valdymo sistemos (TVS)* [interaktyvus]. 2008, nr 10 [Žiūrėta 2016 m. Birželio 12 d.]. Prieiga per internetą: http://www.softconsulting.lt/straipsniai/turinio_valdymo_sistemos_tvs/
8. Yoshitaka, S. *WebDevelopment 101: Top Web Development Languages in 2014* [interaktyvus]. 2014 [Žiūrėta 2016 m. Birželio 1 d.]. Prieiga per internetą: <https://www.upwork.com/blog/2014/03/web-development-101-top-web-development-languages-2014/>.
9. JAYARATHE, G. *MySQL Basics with PHP* [interaktyvus]. 2013 [Žiūrėta 2017 m. Balandžio 3 d.]. Prieiga per internetą: <http://samples.leanpub.com/mysqlbasics-sample.pdf>
10. Jiang, L., Rabah, S., Mingzhi, L. *Comparative studies of 10 programing languages within 10 diverse criteria* [interaktyvus]. Concordia universitetas [Montreal, Canada], 2009 [Žiūrėta 2017 m. Kovo 7 d.]. Prieiga per internetą: <https://arxiv.org/ftp/arxiv/papers/1009/1009.0305.pdf>
11. Lynch, J. P., Horton, S. *Site structure* [interaktyvus]. [žiūrėta 2016 m. Gegužės 29 d.]. Prieiga per internetą: <http://webstyleguide.com/wsg3/3-information-architecture/3-site-structure.html>.
12. New relic. *PHP* [interaktyvus]. 2011 [Žiūrėta 2017 m. Kovo 3 d.]. Prieiga per internetą: <http://www.seas.es/blog/wp-content/uploads/2013/02/php368.png>
13. Python. *Python 3.6.1 documentation* [interaktyvus]. 2017 [Žiūrėta 2017 m. Kovo 23 d.]. Prieiga per internetą: <https://docs.python.org/3>
14. Purer, K. *PHP vs. Python vs. Ruby – The web scripting language shootout* [interaktyvus]. Viena universitetas [Viena, Austria]. 2009 [Žiūrėta 2017 m. Kovo 26 d.]. Prieiga per internetą: <https://klau.si/files/php-vs-python-vs-ruby.pdf>
15. Ruby. *Documentation* [interaktyvus]. 2017 [Žiūrėta 2017 m. Kovo 23 d.]. Prieiga per internetą: <https://www.ruby-lang.org/en/documentation/>
16. Skill value. *Time to learn – Ruby programing language* [interaktyvus]. 2015 [Žiūrėta 2017 m. Kovo 7 d.]. Prieiga per internetą: <http://blog.ittestsonline.com/ruby-2/time-to-learn-ruby-programming-language/>
17. Tutorials point. *Python overview* [interaktyvus]. 2016 [Žiūrėta 2017 m. Kovo 28 d.]. Prieiga per internetą: https://www.tutorialspoint.com/python/python_overview.htm
18. Valiukėnas, V., Sabaliauskaitė, R., Gabijūnienė, R. *Aiškinamasis telekomunikacijų terminų žodynas*. Vilnius: Lietuvos telekomas, 2004.

19. Visart. *Interneto svetainė*. Kas yra interneto svetainė? [interaktyvus]. [Žiūrėta 2016 m. Gegužės 28 d.]. Prieiga per internetą: <http://www.visart.lt/interneto-svetaine>.
20. W3. *HTML & CSS* [interaktyvus]. [Žiūrėta 2016 m. Birželio 12 d.]. Prieiga per internetą: <https://www.w3.org/standards/webdesign/htmlcss>
21. Wamp server. *Start with wampserver* [interaktyvus]. 2004 [Žiūrėta 2017 m. Balandžio 3 d.]. Prieiga per internetą: <http://www.wampserver.com/en/>
22. ŽEBRAUSKAS, S., JANUŠEVIČIUS, V., MARCINKEVIČIUS. *Aiškinamasis elektrotechnikos terminų žodynas : lietuvių k., vokiečių k., anglų k., prancūzų k., rusų k.* Kaunas : Technologija, 2010.

PRIEDAI

1 PRIEDAS

PHP interneto svetainės failo `webPHP.php` programavimo kodas:

```
<?php
require_once 'init.php';
$query = $db->prepare("
    SELECT id, name, done, created
    FROM items
");
$query->execute();
$items = $query->rowCount() ? $query : [];
?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>PHP Website</title>
        <link href="https://fonts.googleapis.com/css?family=Open+Sans
rel="stylesheet">
        <link href="https://fonts.googleapis.com/css?family=Russo+One
rel="stylesheet">
        <link rel="stylesheet" href="../maincss.css">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <div class="list">
            <h1 class="header">To do list</h1>
            <?php if(!empty($items)): ?>
            <ul class="items">
                <?php foreach($items as $item): ?>
                <li>
                    <span class="item <?php echo $item['done'] ? ' done ' :
' ' ?>><?php echo $item['name']; ?></span>
                    <div class="container">
                        <a href="delete.php?item=<?php echo $item['id']; ?>"
class="done-button">Delete</a>
                        <span class="done-button"> <?php echo $item['created'];
?> </span>
                        <?php if(!$item['done']): ?>
                        <a href="mark.php?as=done&item=<?php echo $item['id'];
?>" class="done-button">Mark as done</a>
                        <?php endif; ?>
                    </div>
                </li>
                <?php endforeach; ?>
            </ul>
            <?php else: ?>
                <p>You haven't added any items yet.</p>
            <?php endif; ?>
            <form class="item-add" action="add.php" method="post">
                <input type="text" name="name" placeholder="Type a new item
here." class="input" autocomplete="off" required>
                <input type="submit" value="Add" class="submit">
            </form>
        </body>
```

</html>

2 PRIEDAS

PHP interneto svetainės failo `add.php` programavimo kodas:

```
<?php
require_once 'init.php';
if(isset($_POST['name'])) {
    $name = trim($_POST['name']);
    if(!empty($name)) {
        $addedQuery = $db->prepare("
            INSERT INTO items (name, done, created)
            VALUES (:name, 0, NOW())
        ");
        $addedQuery->execute([
            'name' => $name
        ]);
    }
}
header('Location: webPHP.php');
```

3 PRIEDAS

PHP interneto svetainės failo `delete.php` programavimo kodas:

```
<?php
require_once 'init.php';
if(isset($_GET['item'])) {
    $id = $_GET['item'];
    if(!empty($id)) {
        $deleteQuery = $db->prepare("
            DELETE
            FROM items
            WHERE id=:id
        ");
        $deleteQuery->execute([
            'id' => $id
        ]);
    }
}
header('Location:webPHP.php');
```

4 PRIEDAS

PHP interneto svetainės failo `init.php` programavimo kodas:

```
<?php
error_reporting(E_ALL);
session_start();
$db = new PDO('mysql:dbname=todolist;host=127.0.0.1', 'root', '');

```

5 PRIEDAS

PHP interneto svetainės failo `mark.php` programavimo kodas:

```
<?php
require_once 'init.php';
if(isset($_GET['as'], $_GET['item'])) {
    $as      = $_GET['as'];
    $item = $_GET['item'];

    if($as) {
        $doneQuery = $db->prepare("
            UPDATE items
            SET done = 1
            WHERE id = :item
        ");
        $doneQuery->execute([
            'item' => $item
        ]);
    }
};
header('Location: webPHP.php');
```

6 PRIEDAS

PHP, Python ir **Ruby** interneto svetainės failo `main.css` programavimo kodas:

```
body {
    background-color:#eceff0;
}

body,
input {
    font:1em "Open Sans", sans-serif;
    color:#363639;
}

a {
    text-decoration:none;
    border-bottom:1px dashed #363639;
}

/* List */
.list {
    background-color:#fff;
    margin: 20px auto;
    width: 100%;
    max-width:500px;
    padding:20px;
    border-radius:10px;
    box-shadow:3px 3px 0 rgba(0, 0, 0, .1);
    box-sizing:border-box;
}

.list .header {
    font-family: "Russo One", cursive;
    margin: 0 0 10px 0;
    text-align:center;
    border-bottom:3px dashed #363639;
}
```

```
/* Items */
.item {
    word-wrap:break-word;
}
.items {
    margin:0;
    padding:0;
    list-style-type:none;
}
.items li:hover .done-button{
    opacity:1;
}
.items .item.done {
    text-decoration:line-through;
}
.items li,
.item-add .input {
    border:0;
    border-bottom:1px dashed #ccc;
    padding: 20px 0;
}
.input {
    width:100%;
}
.input:focus {
    outline:none;
}
/* Done button */
.done-button {
    display:inline-block;
    font-size: 0.8em;
    background-color:#d9dfef;
    color:#363639;
    padding: 1px 3px;
    margin:1px;
    border:0;
    opacity:0.4;
    float: right;
}
/* Submit button */
.submit {
    background-color:#fff;
    padding:5px 10px;
    border:1px solid #ddd;
    width:100%;
    margin-top:10px;
    box-shadow: 3px 3px 0 #ddd;
}
.container {
}
}
```

Python interneto svetainės failo webPython.py programavimo kodas:

```
#!/c:/Python35/python.exe
import cgi, cgi, pymysql
import init

form = cgi.FieldStorage()
a = init.conn.cursor()
sql = 'SELECT id, name, done, created FROM `items`'
a.execute(sql)
rows = a.fetchall()
print ("Content-type:text/html\r\n\r\n")
print ("<!DOCTYPE html>")
print ("<html lang='en'>")
print ("<head>")
print ("<meta charset='UTF-8'>")
print ("<title>Python Website</title>")
print ("<link href='https://fonts.googleapis.com/css?family=Open+Sans'
rel='stylesheet'>")
print ("<link href='https://fonts.googleapis.com/css?family=Russo+One'
rel='stylesheet'>")
print ("<link rel='stylesheet' href='../maincss.css'>")
print ("<meta name='viewport' content='width=device-width, initial-scale=1.0'>")
print ("</head>")
print ("<body>")
print ("<div class='list'>")
print ("<h1 class='header'>To do list</h1>")
if a.rowcount > 0:
    print ("<ul class='items'>")
    for x in rows:
        print ("<li>")
        print ("<span class='item {}'>{}".format('done' if x[2]==1 else
'',x[1]))
        print ("<div class = 'container'>")
        print ("<a href='delete.py?id={}' class='done-
button'>Delete</a>".format(x[0]))
        print ("<span class='done-button'>{}".format(x[3]))
        if not x[2]:
            print ("<a href='mark.py?as=done&item={}' class='done-button'>Mark
as done</a>".format(x[0]))
        print ("</div>")
        print ("</li>")
    print ("</ul>")
else:
    print("<p>You haven't added any items yet.</p>")
print ("<form class='item-add' action='add.py' method='post'>")
print ("<input type='text' name='name' placeholder='Type a new item here.'
class='input' autocomplete='off' required>")
print ("<input type='submit' value='Add' class='submit'>")
print ("</form>")
print ("</div>")
print ("</body>")
print ("</html>")
```

8 PRIEDAS

Python interneto svetainės failo `add.py` programavimo kodas:

```
#!/c:/Python35/python.exe
import pymysql, cgi, cgiib, datetime, init
form = cgi.FieldStorage()
a = init.conn.cursor()

if form.getvalue('name'):
    name = (form.getvalue('name'))
    sql = "INSERT INTO items (name, done, created) VALUES (\'%s\', 0,
CURDATE())"
    a.execute(sql%(name))
print ("Location: webPython.py\r\n")
print ("Connection: close \r\n")
print ("")
```

9 PRIEDAS

Python interneto svetainės failo `delete.py` programavimo kodas:

```
#!/c:/Python35/python.exe
import pymysql, cgi, cgiib, init
form = cgi.FieldStorage()
a = init.conn.cursor()

if form.getvalue('id'):
    id = form.getvalue('id')
    sql = "DELETE FROM items WHERE id={}".format(id)
    a.execute(sql)
print ("Location: webPython.py\r\n")
print ("Connection: close \r\n")
print ("")
```

10 PRIEDAS

Python interneto svetainės failo `init.py` programavimo kodas:

```
#!/c:/Python35/python.exe
import pymysql, cgi, cgiib
cgiib.enable()
conn = pymysql.connect(host='localhost', user='root', passwd='', db='todolist')
```

11 PRIEDAS

Python interneto svetainės failo `mark.py` programavimo kodas:

```
#!c:/Python35/python.exe
import pymysql, cgi, cgi, init
form = cgi.FieldStorage()
a = init.conn.cursor()
if form.getvalue('as') and form.getvalue('item'):
    aas = form.getvalue('as')
    item = form.getvalue('item')
    sql = "UPDATE items SET done = 1 WHERE id =%s"
    a.execute(sql%(item))
print ("Location: webPython.py\r\n")
print ("Connection: close \r\n")
print ("")
```

12 PRIEDAS

Ruby interneto svetainės failo `config.ru` programavimo kodas:

```
require "greeter"
use Rack::Reloader, 0

run Rack::Cascade.new([Rack::File.new("public"), Greeter])
```

13 PRIEDAS

Ruby interneto svetainės failo `index.html.erb` programavimo kodas:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Ruby Website</title>
    <link href="https://fonts.googleapis.com/css?family=Open+Sans"
rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Russo+One"
rel="stylesheet">
    <link rel="stylesheet" href="stylesheets/maincss.css" type="text/css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div class="list">
      <h1 class="header">To do list</h1>
      <% if !@item %>
        <ul class="items">
          <% for @item in results %>
            <li>
              <span class="item <%= if @item['done'] == 1 then 'done'
else '' end%>"><%= @item["name"] %></span>
              <div class="container">
                <a href="delete?item=<%= @item["id"] %>" class="done-
button">Delete</a>
                <span class="done-button"><%= @item["created"] %></span>
                <a href="mark?as=done&item=<%= @item["id"] %>"
class="done-button">Mark as done</a>
            </li>
          </%>
        </ul>
      </%>
    </div>
  </body>
</html>
```

```
        </div>
      </li>
    <% end %>
  </ul>
  <% else %>
    <p>You haven't added any items yet.</p>
  <% end %>
  <form class="item-add" action="add" method="post">
    <input type="text" name="name" placeholder="Type a new item
here." class="input" autocomplete="off" required>
    <input type="submit" value="Add" class="submit">
  </form>
</div>
</body>
</html>
```

14 PRIEDAS

Ruby interneto svetainės failo `add.rb` programavimo kodas:

```
require "init"
class Add
  def initialize(req)
    @request = req
  end
  def addLine
    if @request.params["name"] then
      item = @request.params["name"]
      client = Init.new.getClient();
      sql = client.prepare("INSERT INTO items (name, done, created)
VALUES (?, 0, CURDATE())")
      sql.execute(item)
    end
  end
end
```

15 PRIEDAS

Ruby interneto svetainės failo `delete.rb` programavimo kodas:

```
require "init"
class Delete
  def initialize(req)
    @request = req
  end
  def deleteLine
    if @request.params["item"] then
      item = @request.params["item"]
      client = Init.new.getClient();
      sql = client.prepare("DELETE FROM items WHERE id=?")
      sql.execute(item)
    end
  end
end
```

16 PRIEDAS

Ruby interneto svetainės failo `init.rb` programavimo kodas:

```
require "mysql2"
class Init

  def getClient()
    return Mysql2::Client.new(:host => "localhost", :username => "root",
      :password => "", :database => "todolist")
  end

end
```

17 PRIEDAS

Ruby interneto svetainės failo `mark.rb` programavimo kodas:

```
require "init"
class Mark
  def initialize(req)
    @request = req
  end
  def markLine
    if @request.params["as"] and @request.params["item"] then
      as = @request.params["as"]
      item = @request.params["item"]
      client = Init.new.getClient();
      sql = client.prepare("UPDATE items SET done = 1 WHERE id =?")
      sql.execute(item)
    end
  end
end
```

18 PRIEDAS

Ruby interneto svetainės failo `greeter.rb` programavimo kodas:

```
require "erb"
require "init"
require "delete"
require "add"
require "mark"

class Greeter
  def self.call(env)
    new(env).response.finish
  end

  def initialize(env)
    @request = Rack::Request.new(env)
  end

  def response
    case @request.path
    when "/" then Rack::Response.new(render("index.html.erb"))
    when "/delete" then
      del=Delete.new(@request)
      del.deleteLine();
    end
  end
end
```

```
        Rack::Response.new(render("index.html.erb"))
    when "/add" then
        add=Add.new(@request)
        add.addLine();
        Rack::Response.new(render("index.html.erb"))
    when "/mark" then
        mark=Mark.new(@request)
        mark.markLine();
        Rack::Response.new(render("index.html.erb"))
    else Rack::Response.new("Not Found", 404)
    end
end
def render(template)

    client=Init.new.getClient();
    results = client.query("SELECT id, name, done, created FROM
items").to_a
    path = File.expand_path("../views/#{template}", __FILE__)
    ERB.new(File.read(path)).result(binding)
end
end
```