

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Bakalauro baigiamasis darbas

Internetinių puslapių generavimas

(Web Page Generation)

Atliko: 4 kurso, 1 grupės studentas

Lukas Tyla (parašas)

Darbo vadovas:

Irmantas Radavičius (parašas)

Vilnius

2015

Turinys

Ivadas	3
1. Dinaminiai ir statiniai internetiniai puslapiai	7
1.1. Dinaminiai internetiniai puslapiai.....	7
1.2. Statiniai internetiniai puslapiai.....	9
1.3. Kokybiško internetinio puslapio kriterijai.....	9
1.4. Skirtumai tarp dinaminių ir statinių internetinių puslapių.....	11
2. Statinių internetinių puslapių generavimas.....	16
2.1. Statinių internetinių puslapių generatorius	16
2.2. Statinių internetinių puslapių generavimo nauda.....	17
2.3. Populiariausias „Jekyll“ generavimo įrankis	19
3. Statinis internetinių puslapių generavimas	23
3.1. Dinaminio internetinio puslapio konversija į statinį	23
3.2. Statinio internetinio puslapio konversija į dinaminį.....	25
3.3. Dinaminio puslapio perdarymo į statinį eksperimentas	26
3.3.1. Dinaminio puslapio analizė	27
3.3.2. Dinaminio puslapio konversijos eiga.....	29
4. Eksperimento analizė	31
Rezultatai ir išvados	36
Literatūros sąrašas	39

Išvadas

Internetiniai puslapiai – tai pati populiariausia terpė skleisti ir dalintis informacija visame pasaulyje. Šiuo metu suskaičiuojama virš 1 milijardo registruotų kompiuterių adresų, tačiau tik apie 25% iš jų yra aktyviai naudojami internetiniams puslapiams. [Int16] Nemenką dalį registruotų internetinių puslapių sudaro adresų pardavimo sistemose įregistruoti, bet niekur nenaudojami puslapiai, dažniausiai skirti adresų per pardavinėjimui, populiariausia registro ir internetinių puslapių talpinimo sistema „GoDaddy“ 2016 m sausį turėjo virš 61 milijono registruotų adresų sistemoje. [Wik16]

Spartus internetinių puslapių tobulėjimas kartu su technologinėmis inovacijomis interneto prieinamumo ir spartos bei mobiliųjų technologijų srityse lėmė labai greitą internetinių puslapių populiarumo šuolį ir platų panaudojimo spektrą. Kas prieš dešimt metų atrodė neįmanoma, dabar tapo daugelio kasdienybe. Seniau tik nepatogiam naršymui mobiliems įrenginiams nepritaikytuose puslapiuose naudoti telefonai be skaičiuojamųjų pajėgumų susidoroti su sudėtingais puslapiais plačiai naudojančiais „JavaScript“ kalbą šiandien tapo pagrindine duomenų sklaidos ir pasiekimo platforma. 2014 metais mobiliuosius įrenginius turinčių naudotojų skaičius viršijo stacionarius įrenginius turinčiuosius ir pakeitė daugelio požiūrį į internetinius puslapius. [Cha16] Google skatina internetinių puslapių savininkus adaptuoti jų puslapius patogesniai naudojimui mobiliuose įrenginiuose suteikdami prioritetą mobiliam įrenginiui optimizuotą internetinį puslapį paieškos rezultatuose pateikti aukščiau, kuomet paieška naudojama mobiliuoju įrenginiu. [Goo16] Šie pasikeitimai lėmė ir žmonių įpročių pakitimus, daugelis naudoja net keletą socialinių tinklų, naujienų portalų, muzikos ir vaizdo įrašų sklaidos portalų, enciklopedijų, skaičiavimo sistemų.

Platus kiekis sistemų ir daugybė naudotojų sukuria milžinišką duomenų srautą keliaujantį internetu, vos per minutę pasaulyje 2015 metais vidutiniškai išsiųsti apie 205 milijonai elektroninių laiškų, apie 44 milijonai susirašinėjimo sistemos „WhatsApp“ žinučių, įkelti apie 400 valandų „Youtube“ vaizdo įrašų, parašyti apie 3,3 milijono „Facebook“ įrašų bei atliktos apie 3,1 milijono „Google“ užklausos. [All16] Toks masyvus kiekis internetu keliaujančių duomenų srauto sukuria poreikį vis galingesniems serveriams, galintiems susidoroti su tokiu kiekiu naudotojų ir duomenų per trumpą laiką. Nuolatinis duomenų kiekio didėjimas rodo, jog

apkrovos serveriams tik augs, o internetiniuose puslapiuose atliekamų užduočių kiekis ir sudėtingumas kils.

Besikeičiantys žmonių įpročiai ir poreikiai bei technologijų galimybės atveria naujus poreikius ir naujas galimybes daugelyje sferų. Internetiniai puslapiai – viena iš daugiausiai pokyčių išgyvenusių rinkų. Seniau, tik paprastų dokumentų keitimuisi naudota sistema, dabar įgalina naudotojus atlikti daugybę skirtingų darbų. Tapo natūralu tikėtis, kad naudojantis internetiniais puslapiais galime skaityti ir rašyti elektroninius laiškus bei žinutes, klausytis muzikos, žiūrėti vaizdo įrašus, skaityti knygas, ieškoti įvairiausios informacijos, atlikti sudėtingus skaičiavimus, sekti ir bendrauti su daugybe žmonių visame pasaulyje. Platus funkcionalumo spektras internetinius puslapius paverčia itin sudėtingomis sistemomis, kurios dažnai reikalauja pastovaus duomenų keitimosi su duomenų baze, didelių serverio skaičiuojamųjų resursų bei plačios interneto linijos, kad galėtų apdoroti didelį kiekį naudotojų vienu metu. Nėra tobulos programinės įrangos be galimų spartos ir saugumo spragų, todėl ir daugelis, ypač sudėtingesnių, internetinių puslapių turi spartos ir saugumo problemų. Šios problemos tampa vis opesnės didėjant ir populiarėjant internetiniams puslapiams. Sudėtingos sistemos įgalinančios įvairų puslapio funkcionalumą dažniausiai remiasi dinaminio puslapio generavimo būdu, serveris gavęs kiekvieną užklausą atverti tam tikrą puslapį kreipiasi į duomenų bazę reikiamiems duomenims gauti ir remiantis duomenimis generuoja reikiamą puslapio struktūrą su reikiamais elementais ir funkcionalumu realiu laiku. Nesunku pastebėti, kad toks internetinių puslapių pateikimo būdas reikalauja papildomo laiko ir, priklausomai nuo užduoties sudėtingumo – serverio resursų. Dinamiškai generuojant internetinius puslapius yra atveriamos galimybės net menkiausioms klaidoms vienoje iš sisteminių dalių sukelti didelių problemų, o neištaisytos saugumo spragos gali parkludyti visą sistemą „ant kelių“ ar rimtai sutrikdyti jos darbą. Saugumo spragos dažniausiai sudaro palankesnes sąlygas įsibrauti į sistemą ar bent duomenų bazę, suteikiant galimybę perimti sistemos valdymą ar sugadinti arba ištrinti sistemoje esančius duomenis, visiškai sutrikdant sistemos darbą. Neretai sistemoms tenka susidurti su sunkiai įveikiamomis tarnybos neigimo atakomis (angl. Denial of Service). Šios atakos stipriausiai paveikia sudėtingas sistemas, kurioms kiekviena puslapio pateikimo užklausa užtrunka ilgesnį laiko tarpą, tokiu būdu pakanka net nedidelio nuolatinio kiekio užklausų į serverį ir puslapis tampa niekam neprieinamas, nes užklauskos nespėjamos apdoroti.

Matome, jog internetiniai puslapiai yra naudojami itin plačiai ir jų populiarumas tik auga. Pastaraisiais metais minėtieji puslapiai išgyvena pasikeitimus, tampa privalomas mobiliųjų platformų palaikymas, auga puslapių sudėtingumas. Duomenų srautas perduodamas internetu nuolatos didėja ir vis stipriau apkrauna serverius, turinčius apdoroti visus reikiamus duomenis. Augantis sistemų populiarumas ir sudėtingumas atneša ir neigiamų faktorių, didesnė klaidų ir saugumo spragų tikimybė bei ilgesnis užklauso apdorojimo laikas. Šie pokyčiai verčia ieškoti alternatyvių sprendimų spartos ir saugumo problemoms. Galima rasti sprendimų, eliminuojančių daugelį dinaminių internetinių puslapių problemų, vienas jų – statiška generuojami internetiniai puslapiai. Darbe bus nagrinėjamas šio metodo veikimas, privalumai, trūkumai ir kada yra tinkamiausia naudoti generuojamus statinius puslapius.

Tikslas. Šiuo darbu siekiama nustatyti statinių puslapių generavimo panaudojimo galimybes lyginant su dinaminiais puslapiais.

Uždaviniai.

- Apibrėžti, kas yra statinis ir dinaminis puslapis;
- Apibrėžti kokybiško internetinio puslapio kriterijus;
- Palyginti statinius ir dinامينius internetinius puslapius pagal kokybės kriterijus;
- Išskirti statinių ir dinaminių puslapių privalumus ir trūkumus;
- Apibrėžti statinio ir dinaminio puslapio gavimo kelius;
- Suprasti, kas yra statinių internetinių puslapių generavimas ir kokia to nauda;
- Išanalizuoti populiariausio statinių puslapių generatoriaus veikimo principus;
- Apibrėžti statinio ir dinaminio puslapio konversijos metodiką;
- Atlikti dinaminio puslapio konversijos į statinį puslapį eksperimentą;
- Rasti statinių puslapių generavimo privalumus ir trūkumus prieš dinامينius puslapius.

Darbo struktūra. Pirmame darbo skyriuje aiškinamasi ir apibrėžiama, kas yra statinis ir dinaminis internetinis puslapis, jų privalumai ir trūkumai bei išskiriami kokybiško internetinio puslapio kriterijai. Antrame skyriuje aiškinamasi, kas yra statinių puslapių generatorius, nustatoma jų nauda ir analizuojamas populiariausio (darbo rašymo metu) statinių puslapių generavimo įrankio „Jekyll“ veikimas. Trečiame skyriuje pateikiami konversijos tarp statinio ir

dinaminio internetinio puslapio principai, vykdomas dinaminio puslapio konversijos į statinį puslapį eksperimentas, aiškinami eksperimento rezultatai bei pateikiami siūlymai, kokiose situacijose, kuris puslapis yra tinkamiausias ir teisingiausias pasirinkimas.

1. Dinaminiai ir statiniai internetiniai puslapiai

Internetiniai puslapiai gimė 1989 metais Tim Berners-Lee rankose kaip būdas mokslininkams dalintis aktualiais dokumentais ir atradimais visame pasaulyje. [Cer16] Tuo metu juodomis raidėmis baltame lape parašytam turiniui peržiūrėti naudoti internetiniai puslapiai šiandien sutinkami kardinaliai pasikeitę. Per kiek daugiau nei ketvirtį amžiaus internetiniai puslapiai ne tik tapo pritaikomi daugelyje sričių, „JavaScript“ kalbos pagalba įgavo galimybę būti interaktyvūs – reaguoti į naudotojo veiksmus ne tik persiunčiant į nurodytą puslapį, bet ir geba atrodyti labai įvairiai naudojantis pakopinių stilių taisyklėmis (angl. Cascading Style Sheets). Nenuostabu, kad tokie pasikeitimai leido internetinius puslapius pritaikyti labai plačiai auditorijai, nuo daktaro laipsnį turinčių specialistų iki vaikų. Toks platus puslapių panaudojimo spektras atvėrė galimybes sukurti įvairias sistemas internetiniame puslapyje, tiksliau – internetinius puslapius naudoti kaip interakcijos priemonę su pačiomis sudėtingiausiomis sistemomis, kurios tik puslapiui užsikrovus suteikia plačią funkcijų aibę naudotojams. Internetiniai puslapiai iš statinių dokumentų reprezentacijų nesunkiai tapo pilnai dinaminėmis sistemomis.

1.1. Dinaminiai internetiniai puslapiai

Netrukus po internetinių puslapių pasirodymo buvo pastebėtas poreikis dinaminiam puslapių funkcionalumui, duomenų įterpimui iš duomenų bazių, naudotojų paskyroms, paieškai, komentarams ir pan. Įvairių funkcijų poreikis reikalavo bendravimo su duomenų baze ir funkcijų vykdymo serverio pusėje. Šiam tikslui 1993 metais buvo sukurtas bendrosios tinklų sąsajos (angl. CGI – Common Gateway Interface) standartas nurodęs, kaip turi būti integruojami scenarijai (angl. scripts) ar programos į hiperteksto perdavimo protokolo serverius (angl. HTTP - Hyper Text Transfer Protocol). [Wik16a] Šis protokolas atvėrė kelią tolimesniems darbas su dinaminiais internetiniais puslapiais ir per sekančius keturis metus buvo išleistos pagrindinės, iki šiol aktyviai naudojamos programavimo kalbos ar jų plėtiniai skirti naudoti kaip internetinių puslapių serveriai, tai yra Perl, Python PHP, Ruby, Cold Fusion, ASP ir Java Servlet. [Pin07] Remiantis W3Techs duomenimis, šiuo metu serverių

programavimo kalbų rinką dominuoja PHP kalba naudojama 82,2% iš 10 milijonų populiariausių internetinių puslapių pasaulyje, antroje vietoje su 15,8% rinkos kompanijos „Microsoft“ vystomas ASP.NET karkasas (angl. framework), o trečia pagal populiarumą – Java programavimo kalba. [W3t16] Tokiam PHP kalbos populiarumui didžiausią įtaką daro plačiausiai naudojamos turinio valdymo sistemos (angl. CMS – Content Management System), kaip kad Wordpress, Joomla, Drupal, kurios, atsižvelgiant į W3Techs duomenis 2016 metų gegužės 11 dieną užima apie 70% rinkos kartu sudėjus ir visos minėtos turinio valdymo sistemos yra parašytos naudojantis PHP programavimo kalba. [W3t16a]

Dažniausiai naudojami dinaminiai internetiniai puslapiai, kurių sukūrimas valdomas serveryje dirbančia programine įranga. Tai puslapiai, kurių turinys nėra nustatytas ir žinomas prieš atidarant juos. Puslapių atidarymo metu gali būti vykdomi įvairūs tikrinimai su duomenų baze, išoriniais serveriais ar kita informacija. Priklausomai nuo to, kokios sąlygos bus tenkinamos, atitinkamai skirtingas puslapis bus sukuriamas kiekvieną kartą jį atidarant. Toks procesas leidžia sukurti pilnai personalizuotus internetinius puslapius ar sistemas, kuriose kiekvienas naudotojas turi savo erdvę, mato jam aktualų turinį, gali reikšti savo nuomonę komentaruose ar rašydamas įrašus, ieškoti turinio puslapiuose pagal savo sugalvotus užklausos raktažodžius. Dinaminiai puslapiai taipogi leidžia paslėpti ar rodyti reikiamą turinį priklausomai nuo prisijungusio naudotojo, įgalina balsavimą, diskusijų forumus, elektroninių parduotuvių funkcionalumą ar bet kokią kitą interaktyvų veiksmą, kurio veikimą galima sukurti naudojantis programavimo kalbomis.

Dinaminių internetinių puslapių populiarumą lėmė jų lankstumas, galimybė lengvai keisti turinį internetiniame puslapyje, net neturint techninių žinių, nes teksto rašymas sistemose dažniausiai pateikiamas patogiam ir aiškiam tekstiniame redaktoriuje, kuriame rašomas turinys atitinka turinį, kurį bus gaunamas jį išsaugojus, tokie redaktoriai dažnai vadinami KMTIB – Kaip Matai Taip ir Bus (angl. WYSIWYG – What You See Is What You Get).

1.2. Statiniai internetiniai puslapiai

Pirmasis internetinis puslapis buvo statinis ir bent apie keturis metus po to sukurti internetiniai puslapiai turėjo būti statiniai, nes struktūra kurti dinامينius internetinius puslapius pradėta plačiau vystyti tik 1993 metais, kuomet buvo paskelbtas bendras standartas, nurodantis, kaip internetiniai puslapiai turi bendrauti su serveriu. [Wik16a] Statiniai internetiniai puslapiai yra paprasčiausi dokumentai suformuoti naudojantis HTML kalba ir saugomi serveryje. Naudotojui atlikus užklausą į statinį puslapį – jam tiesiog atsiunčiamas serveryje esantis failas su kitais reikalingais resursais ir naudotojo naršyklėje atvaizduojamas internetinio puslapio turinys. Toks sprendimas leidžia statinius internetinius puslapius atvaizduoti tokius pačius kiekvienam naudotojui, nepriklausomai nuo jokių parametrų.

1.3. Kokybiško internetinio puslapio kriterijai

Internetinių puslapių gausa internete gali stebinti, bet populiariausi ir labiausiai lankomi - kokybiški puslapiai. Tikėtina, kad naudotojas apsilankęs patogiam, kokybišką bei aktualų turinį teikiančiame puslapyje, jo kokybę atsimins ir vėliau. Kiekvienas sutinkamas puslapis nebūtinai yra kokybiškas, todėl turėtų būti būdas nustatyti internetinio puslapio kokybę. Šią problemą pastebėjo V. S. Moustakis, C. Litos, A. Dalivigas, L. Tsironis ir atlikę tyrimą apibrėžė devynis internetinio puslapio kokybės kriterijus. [MLD04] Anot autorių atliktos apklausos ir tyrimo, jie svarbumo mažėjimo tvarka pateikė devynis kriterijus (skliausteliuose pateiktas jų tyrime nustatytas svarbumas): aktualumas (18%), naudingumas (15%), patikimumas (9%), specializacija (9%), architektūra (9%), navigacijos aiškumas (7%), efektyvumas (6%), išdėstymas (angl. layout) (5%), animacija (3%). Lentelėje 1 pateikiami kriterijai ir jų apibrėžimai pagal straipsnio autorius.

1 lentelė. Kokybiško internetinio puslapio kriterijai [MLD04]

Kriterijus	Apibrėžimas
Aktualumas	Lankytoji sukeltas aktualumo ir kokybės jausmas pagal jo interesus. Internetinis puslapis tiek aktualus, kiek aktuali informacija pateikiama naudotojui.
Naudingumas	Naudingumas koreliuoja su aktualumu, jis nusako, ar naudotojas randa puslapyje tokią informaciją, kokios ieškojo. Puslapyje esanti informacija gali būti apie susijusi su ieškomu kontekstu, bet neatsakyti į ieškančiojo klausimą.
Patikimumas	Patikimumas koreliuoja su tikslumu ir nurodo, jog puslapyje esanti informacija yra tiksli ir atnaujinta, dažnai informacijos naujumui parodyti puslapyje yra pateikiama paskutinio atnaujinimo data.
Specializacija	Specializacija apibūdina informacijos puslapyje detalumą, aiškų situacijos pateikimą, pasako, jog puslapyje turi būti visa informacija, kurios lankytoji gali prireikti.
Architektūra	Architektūra apibrėžia puslapio objektų struktūrą, tai, kaip jie perteikia informaciją. Spalvos, specialūs efektais ar mygtukų išdėstymas priklauso architektūros kriterijui.
Navigacijos aiškumas	Navigacijos aiškumas apibrėžia navigacijos puslapyje lengvumą ir patogumą.
Efektyvumas	Efektyvumas vertina puslapio spartą, užsikrovimo greitį, nurodo ar puslapis yra greitai reaguojantis į naudotojo veiksmus.
Išdėstymas	Išdėstymas nusako objektų unikalaus pristatymo egzistavimą puslapyje, kokioje formoje pateikiamas turinys.
Animacija	Animacijos kriterijus kalba apie judančius atvaizduojamo turinio aspektus.

Internetinių puslapių kokybės kriterijų tyrimas atliktas kiek daugiau nei prieš dešimtmetį, o technologijų rinka tobulėja itin sparčiai, o išvados atliktos pasinaudojant pakankamai nedidele (122 žmonių) ir homogeniška (studentai) imtimi, todėl atsižvelgti į pateiktus kriterijų svorius nėra teisinga. Šiandien, pagrindinė internetinių puslapių naršymo platformai

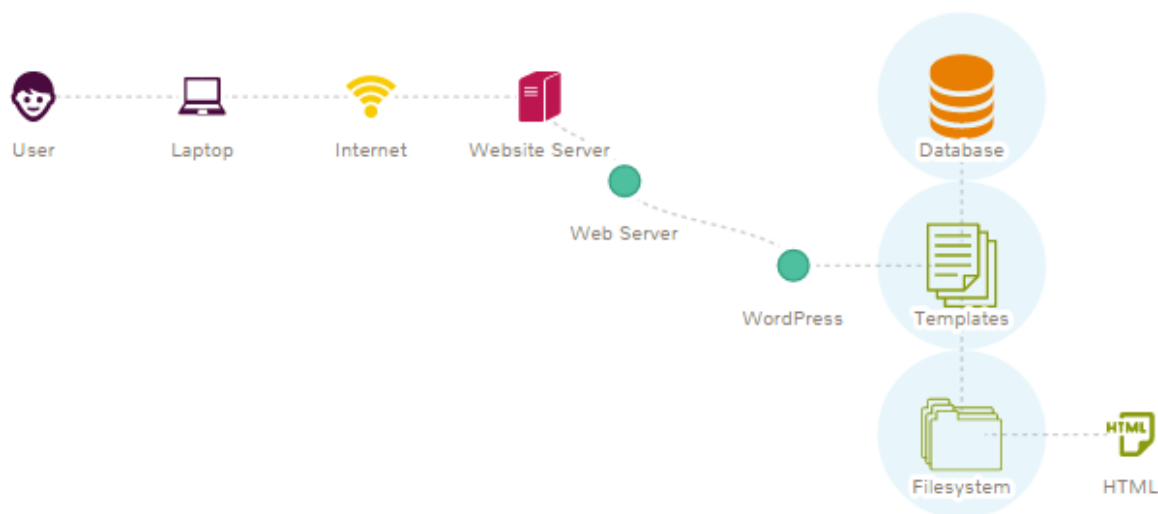
pasikeitus į mobiliąją, efektyvumas tapo kur kas svarbesnis, nes mobiliųjų įrenginių skaičiuojamoji galia ir naršyklių galimybės nusileidžia stacionarių kompiuterių sugebėjimams, todėl puslapiai turi būti nemažiau funkcionalūs nei seniau, veikti sklandžiai silpnėsiuose įrenginiuose bei būti architektūriškai tinkamais mobiliems įrenginiams. Atsižvelgus į šiandienos internetinių puslapių bei juos naršomų įrenginių tendencijas, efektyvumo ir architektūros kriterijai vertinant internetinių puslapių kokybę turėtų būti aukšto svorio. Galima teigti, kad dėl didėjančio mobiliųjų įrenginių populiarumo, efektyvumo ir architektūros kriterijų svoris turėtų būti didesnis nei aktualumo ir naudingumo, nes net pats naudingiausias ir aktualiausias puslapis gali netenkinti lankytojo, jei jo veikimo ar užsikrovimo sparta jo netenkins arba puslapis bus architektūriškai neoptimizuotas veikti mobiliame įrenginyje.

Minėti devyni internetinio puslapio kokybės vertinimo kriterijai yra aktualūs vertinant puslapio kokybę, bet šiuo darbu siekiama įsitikinti, kiek kokybiškas internetinis puslapis lieka po dinaminio puslapio pavertimo statiniu ar statinio pavertimo diniminiu. Darbo kontekste galima ignoruoti daugumą kriterijų, nes teisingai atlikus konversiją, turinys ar puslapio struktūra ir išvaizda turėtų nepakisti, todėl aktualumo, naudingumo, patikimumo, specializacijos, architektūros, navigacijos aiškumo, išdėstymo ir animacijos kriterijai tampa neaktualūs. Vienintelis kriterijus, kuris lieka aktualus lyginant statinių ir dinaminių puslapių kokybę po konversijos – efektyvumas, šis aspektas ypatingai svarbus atsižvelgiant į rinkoje naudojamų įrenginių, skirtų pasiekti internetinius puslapius, pasiskirstymą.

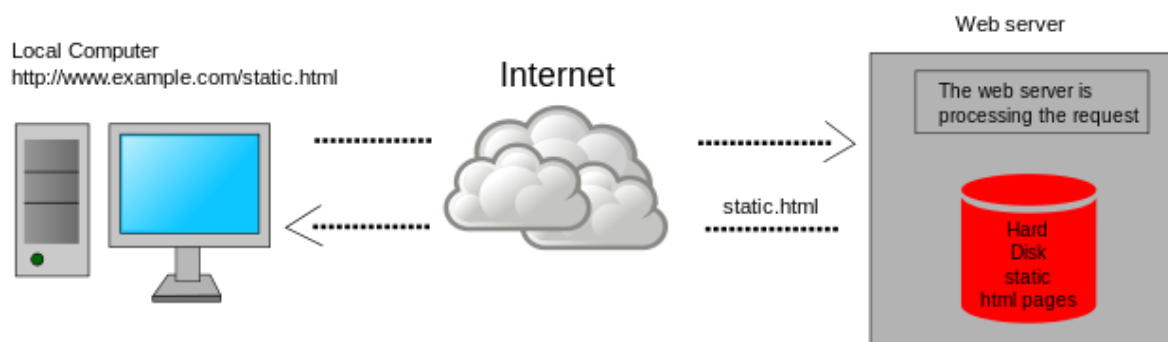
1.4. Skirtumai tarp dinaminių ir statinių internetinių puslapių

Dinaminiai ir statiniai internetiniai puslapiai skiriasi fundamentaliai bei turi skirtingas panaudojimo sritis. Dinaminiai puslapiai neegzistuoja serveryje, jie yra sukuriami realiu laiku, kiekvienos užklauskos į puslapį metu, remiantis įvairia informacija šie puslapiai kiekvieną kartą gali atrodyti skirtingai. Neprisijungęs naudotojas gali matyti tik prisijungimo langą ar dalį turinio, naujam lankytojiui gali būti siūloma registracija sistemoje ar naujienlaiškio (angl. newsletter) prenumerata. Tuo tarpu, visi statiniai internetiniai puslapiai egzistuoja serveryje bet kuriuo laiko metu ir yra pasiekiami bei atrodo taip pat visiems naudotojams, kad ir kada ir iš kur jie prisijungtų.

Pagrindiniai skirtumai tarp statinių ir dinaminių internetinių puslapių siejami su sparta ir saugumu. Statinių puslapių pastovus egzistavimas serveryje leidžia pasiekti didžiausią puslapių atsiuntimo spartą, nes naudotojui atidarant puslapį jam tereikia atsisiųsti serveryje esančio failo kopiją be jokių papildomų duomenų bazės užklausų ar duomenų apdorojimo serverio pusėje. Pavyzdys pateiktas 1 paveiksle parodo detalesnį dinaminio internetinio puslapio gavimo kelią. Šiame paveiksle serveryje naudojama „Wordpress“ turinio valdymo sistema, bet ta pati logika yra pritaikoma bet kokiai turinio valdymo sistemai. Naudotojas kompiuteriu, prijungtu prie interneto, naudodamasis naršykle, nori pamatyti tariamą dinaminį puslapį adresu <http://www.tariamas-dinaminis-puslapis.lt>, tuomet naršyklė siunčia HTTP protokolo GET tipo užklausą į nurodytą interneto serverį, serveryje veikianti turinio valdymo sistema atpažįsta užklausą, randa reikiamus šablonų failus, atlieka aprašytas duomenų bazės užklausas, gavusi atsakymus iš duomenų bazės ir apdorojusi kitus reikiamus duomenis sistema suformuoja HTML puslapį, kuris siunčiamas atgal naudotojui. Tam, kad naudotojas pamatytų internetinį puslapį, atliekama daug veiksmų, šie veiksmas gali užtrukti pakankamai ilgą laiką, ypač tuomet, kai serveris arba duomenų bazė yra stipriai apkraunami įvairių užklausų. Tuo tarpu, naudojantis 2 paveiksle esančia schema, galima pastebėti, jog kelias, iki statinio internetinio puslapio gavimo naudotojui yra kur kas lengvesnis. Statinio puslapio atveju, naudotojas kompiuteriu, prijungtu prie interneto, naudodamasis naršykle, nori atverti tariamą statinį puslapį adresu <http://www.tariamas-statinis-puslapis.lt>, tuomet naršyklė siunčia HTTP protokolo GET tipo užklausą į nurodytą serverį, serveris apdoroja užklausą, failų talpykloje suranda reikiamą HTML failą ir išsiunčia jį atgal naudotojui.



1 pav. Dinaminio internetinio puslapio gavimo kelias. Pagal: <http://nilclass.com/courses/what-is-a-static-website/>



2 pav. Statinio internetinio puslapio gavimo kelias. Pagal: By GgiaEsquema-proxy-internet.svg: Randomicc - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=19280407>

Naudojantis paveiksluose 1 ir 2 pateikta informacija, galima matyti, kad dinaminio puslapio gavimo kelias yra kur kas sudėtingesnis ir laikas, per kurį puslapis parodomas naudotojui gali būti daug didesnis, nei statinio puslapio atveju. Todėl galima teigti, kad statiniai internetiniai puslapiai turi bent minimalią spartos persvarą prieš dinامينius, kuri gali išaugti iki itin didelio skirtumo, kuomet serveris yra stipriai apkraunamas arba atliekama daug užklausų ar skaičiavimų.

Sistemos sudėtingumas dažniausiai tiesiogiai susijęs su jos saugumu, dinaminiai puslapiai nėra išimtis. Juose kur kas daugiau veiklos ir atskirų dalių, kurios turi bendrauti ir keistis informacija vardan būtų sukurtas rezultatas užklausiai. Potencialiai visi dinaminio puslapio sukūrimo kelio žingsniai gali turėti saugumo spragų. Jau pats pirmasis – naudotojo prisijungimas, šiame žingsnyje dažniausiai prašoma pateikti naudotojo vardą ir slaptažodį. Naudotojui įvedus prisijungimo vardą ir slaptažodį jie yra siunčiami į internetinio puslapio serverį, tačiau šiuos siunčiamus duomenis galima perskaityti iš išorės ir taip pasisavinti naudotojo paskyrą, jei siunčiami duomenys nėra tinkamai užšifruojami. Kuomet užšifruoti duomenys saugiai nukeliauja į serverį ir naudotojas yra prijungiamas prie sistemos, saugumo grėsmė gali grėsti turinio valdymo sistemoje, 2 paveikslo atveju – Wordpress, pasak T. Ewer, sistema gali turėti saugumo spragų savo programiniame kode, kurias gamintojas dažnai taiso, tačiau nėra jokios pilnai saugios programinės įrangos ir bet kada aptikta ir neištaisyta saugumo spraga sistemoje gali leisti įsilaužėliui perimti sistemos valdymą ar sugadinti duomenis. [Ewe13]

Pati svarbiausia ir kartu trapiusia dinaminio internetinio puslapio dalis – duomenų bazė, joje saugoma viskas, nuo naudotojų ir jų asmeninių žinučių bei prisijungimų iki kiekvieno įrašo, komentaro ar sisteminio pakeitimo. Tokia svarbi informacija privalo būti saugi, tačiau esant

sisteminėms spragoms galimi įsilaužimai į duomenų bazę, įrašų sugadinimas ar panaikinimas. Dažniausiai duomenų bazę ištinkantys įsilaužimai – SQL injekcijos, tai atvejai, kuomet kurioje nors vietoje formuojant užklausą į duomenų bazę nėra imamasi visų saugumo reikalavimų ir tuomet įsilaužėlis gali įvykdyti bet kokią norimą užklausą. Tokio lygio pakenkimas sistemai gali visiškai sustabdyti internetinio puslapio veikimą ir suteikia galimybę paviešinti svarbią naudotojų informaciją.

Internetiniai puslapiai – visiems naudotojams prieinamas resursas, todėl paslaugos neigimo (angl. Denial of Service) atakos yra dažnai pasitaikančios, nes įsilaužėliai pasitelkę didelius kompiuterių tinklus gali visiškai sustabdyti internetinio puslapio veiklą vykdydami daugybę pasikartojančių užklausų į internetinį puslapį, taip apkraudami serverį ir duomenų bazę tiek, kad sistema nebėra pasiekama niekam kitam, nes nebėra susidoroti su esančiomis užklausomis. [Wik16b] Šios atakos pasekmės tiesiogiai koreliuoja su dinaminio internetinio puslapio sudėtingumu, kuo sudėtingesnis puslapis, kuo daugiau duomenų bazės užklausų ir serverio apdorojimo reikalauja puslapio sukūrimas tuo lengviau paralyžiuoti tokį puslapį pasitelkiant paslaugos neigimo atakas. Nors tokie išpuoliai prieš internetinius puslapius dažniausiai neturi jokių kritinių pasekmių, nes pasibaigus atakai, serveris gali toliau apdoroti reikiamas užklausas, tačiau nuo paskirstytų paslaugos neigimo (angl. Distributed Denial of Service) atakų apsisaugoti praktiškai neįmanoma ir sudėtingesniems puslapiams visada bus sunkiau atsilaikyti nuo tokių išpuolių, o sutrikdytas sistemos veikimas gali neigiamai paveikti dažnai lankomas sistemas, elektronines parduotuves, socialinius tinklus, vaizdo ar muzikos įrašų srautinio duomenų siuntimo (angl. streaming) sistemas. Populiarios sistemos neveiksnumas jūmonei gali pakenkti ne tik sudarant neigiamą įspūdį klientui, bet ir sumažinant pajamas dėl nefunkcionuojančios sistemos.

Didelės dinaminių internetinių puslapių sistemos dažnai turi sudėtingas turinio valdymo sistemas, leidžiančias internetiniame puslapyje koreguoti daugybę parametrų, kurti įrašus, administruoti naudotojus, keisti išvaizdą, kalbą, šablonus ir kita. [W3t16a] Dinaminiai internetiniai puslapiai leidžia turėti plačių galimybių turinio valdymo sistemas, kas naudojantis statiniais internetiniais puslapiais yra neįgyvendinama. Dinaminiai puslapiai įgalina naudotojus bendrauti su sistema, naudojantis komentarais, atsiliepimais, turinio pildymu ar keitimu ir panašiomis funkcijomis.

Sudėtingos dinaminių internetinių puslapių sistemos sukūrimas reikalauja ilgo laiko ir finansų bei galingo serverio sklandžiam sistemos veikimui. Tačiau funkcionali sistema suteikia galimybę lengvai atnaujinti turinį net technologinių sugebėjimų neturinčiam asmeniui.

Galima režiūmuoti, jog dinaminiai ir statiniai internetiniai puslapiai yra pakankamai skirtingi ir pasirinkimas tarp jų nėra intuityviai aiškus. Dinaminiai internetiniai puslapiai suteikia kur kas daugiau galimybių, yra interaktyvūs, leidžia lengvai keisti turinį administruojančiam asmeniui ar reikšti nuomonę lankytoji. Jie suteikia galimybę turėti unikalų ir personalizuotą turinį, saugią naudotojo erdvę bei galimybę turėti sudėtingas informacines sistemas ar teikti įvairias paslaugas internetinio puslapio formoje. Kaip nėra amžinojo variklio, taip neegzistuoja ir pilnai saugi programinė įranga ir kuo kompleksiškesnė PJ, tuo daugiau potencialių saugumo spragų ji turi. Dinaminiai puslapiai, ypač naudojantys populiarias turinio valdymo sistemas, yra dažni taikiniai kibernetinėms atakoms, ypač tuomet, kai aptinkamos naujos saugumo spragos. Dinaminiai internetiniai puslapiai reikalauja sudėtingesnių serverių su papildoma programine įranga bei duomenų bazių, šie reikalavimai atveria daugiau galimybių atsirasti saugumo spragoms ir įvykti kibernetiniam išpuoliui prieš sistemą. Sudėtinga internetinių puslapių sistema leidžia paskirstytoms paslaugos neigimo atakoms lengviau sutrikdyti funkcionavimą dėl ilgesnio užklausos apdorojimo laiko. Todėl statiniai internetiniai puslapiai įgauna persvarą saugumo srityje, nes vienintelė galima įsilaužimo vieta – internetinio puslapio serveris ir net į jį įsilaužus sunku padaryti žalos, nes dažniausiai jame nebus jokios svarbios informacijos, kaip sistemos administratoriaus ar prisijungimo prie duomenų bazės duomenys. Taip pat, statiniai puslapiai suteikia kur kas didesnę spartą dėl paprastesnės struktūros ir nesančio poreikio jokiame duomenų apdorojime serveryje puslapio gavimo metu. Statinių puslapių kūrimas dažniausiai būna pigesnis ir greitesnis bei reikalauja silpnesnių pajėgumų serverio talpinant puslapį. Tačiau sunkus atnaujinimo kelias ir mažesnis interaktyvumo lygis apriboja statinių puslapių panaudojimo galimybes.

2. Statinių internetinių puslapių generavimas

Statiniai internetiniai puslapiai tai puslapiai kuriais prasidėjo interneto era, tai paprasčiausi HTML dokumentai serveryje aprašantys kaip atrodo kiekvienas individualus puslapis interneto svetainėje. [Cer16] Statiniai puslapiai turi didžiausią spartą, mažiausius talpinimo serveryje kaštus ir didžiausią saugumą bei laisvę juos kuriant. Tačiau statiniai internetiniai puslapiai gali sukelti daug problemų, jeigu bandoma sukurti didelės apimties internetinę svetainę naudojantis vien statiniais puslapiais. Kiekvieno atskiro puslapio kūrimas, keitimas ar jo pildymas nauja informacija greitai taptų pačiu nemaloniausiu darbu, jeigu tai reikėtų daryti šimtui ar net tūkstančiui puslapių. Supratus statinių puslapių naudą ir matant poreikį jiems buvo pradėta ieškoti sprendimų masyvių internetinių sistemų kūrimo etapo palengvinimui.

2.1. Statinių internetinių puslapių generatorius

Internetiniai puslapiai būna įvairių tipų ir dydžių, vieni skirti reprezentuoti žmogų ir būna tik vieno puslapio, kiti detalizuoja įmonės veiklą, misiją ir atliktus darbus ir būna sudaryti iš keleto puslapių, neretai sutinkami tinklaraščio (angl. blog) tipo internetiniai puslapiai, kuriuose dažnai talpinami nauji įrašai ir tokia internetinė svetainė gali būti sudaryta iš šimtų ar tūkstančių įrašų. Suprantama, jog vieno puslapio interneto svetainės kūrimas yra pakankamai nesunkus ir intuityvus, bet sukurti tinklaraščio tipo internetinę svetainę sudarytą iš tūkstančių puslapių nėra taip paprasta, greita ir aišku, reikia nepamiršti ir naujų įrašų įterpimo problemos.

Didelių internetinių svetainių kūrimo problemos sprendimui imta ieškoti programinės įrangos darbo palengvinimui. Populiariausi tokios paskirties įrankiai – statinių internetinių puslapių generatoriai. Tai įvairiomis programavimo kalbomis sukurti įrankiai palengvinantys didelių internetinių svetainių raidos procesą ir suteikiantys galimybę nesunkiai atnaujinti puslapius nauju turiniu ar pridėti naujus. Anot E. Bouças, statinių internetinių puslapių generatoriai tinka visiems, tik ne visiems projektams, tai įrankiai, kuriais galima kurti dideles sistemas neapsiribojant įrašais tinklaraščiuose. [Bou15] Visų statinių puslapių generatorių sąrašas pateikiamas kiekvieną dieną statiškai generuojamame <https://www.staticgen.com>

puslapyje. Puslapyje randami įvairiomis kalbomis parašyti generatoriai su nurodytu jų populiarumu (žvaigždučių ir išsišakojimų - atskirų kūrimo atšakų) bei aktyvių klaidų skaičiumi. StaticGen duomenimis 2016 metų gegužės 16 dieną populiariausias įrankis yra „Jekyll“ su daugiau nei 25 tūkstančiais žvaigždučių (skaičius žmonių, kurie suteikė teigiamą įvertinimą projektui), kiti keturi populiarius įrankiai turintys apie dešimt tūkstančių žvaigždučių eilės tvarka yra: „GitBook“, „Hexo“, „Hugo“ bei „Octopress“. [Sta16]

Visų statinių internetinių puslapių generatorių veikimo principas labai panašus. Šie įrankiai veikia panašiai į dinامينius puslapius, jie paima reikalingus duomenis, dažniausiai aprašytus Markdown anotacija ir šablonų pagalba sukuria statinės internetinės svetainės failus, paruoštus talpinimui į serverį. Dažnas generatorius turi galimybę generavimo metu apdoroti išvestinėmis kalbomis sukurtus resursus į standartines kalbas (pvz. CoffeScript į Javascript ar SCSS į CSS). Didžiausias skirtumas tarp įvairių generatorių – šablonams naudojamas variklis (angl. engine), jis nurodo kokia sintaksė naudojama duomenų įterpimui į šablonus bei kokios operacijos su duomenimis palaikomos.

Turinio atskyrimas nuo likusios sistemos dalies suteikia galimybę, pasitelkiant papildomus įrankius, leisti ir naudotojams be techninių žinių kurti, atnaujinti ir naikinti sistemoje esančius duomenis. Dažniausiai statiška generuojamų puslapių kodo talpinimui pasirenkama „GitHub“ platforma dėl lankstumo ir paprastumo, taipogi, šios sistemos siūlomas statinių puslapių talpinimo serveris „GitHub Pages“ naudoja populiariausią generatorių – „Jekyll“.

2.2. Statinių internetinių puslapių generavimo nauda

Statinių internetinių puslapių generatoriai turi daugelį teigiamų savybių. Pasitelkiant generatorius, kur kas lengviau kuriamos sudėtingos ar didelės interneto svetainės, dažniausiai užtenka tik keleto šablonų ir tiek atskirų turinio failų, kiek atskirų puslapių siekiama turėti. Generatoriuose naudojami turinio redagavimo įrankiai supaprastina turinio valdymą ne tik puslapio kūrėjams, bet ir rašytojams bei administratoriams. Yra daugelis žymėjimo kalbų variantų, bet dažniausiai galima sutikti „BBCode“, ypatingai diskusijų forumuose bei „Markdown“, kuri yra plačiausiai paplitusi, labiausiai, dėl naudojimo „GitHub“ sistemoje bei paprasto naudojimo. 3 paveiksle matomas „Markdown“ naudojimo pavyzdys. Galima

pastebėti, kad šia kalba aprašytas tekstas labai artimas generuojamam turiniui, pakanka pridėti norimą kiekį grotelių ir bus sukuriamas tokio lygio antraštė, kiek grotelių buvo nurodyta. Žvaigždutėse įterptas tekstas (pvz.: **Itališkas tekstas**) yra paverčiamas itališko tipo šriftu (pvz.: *Itališkas tekstas*), o dvigubose žvaigždutėse (pvz.: ****ryškus tekstas****) esantis tekstas yra paryškinamas (pvz.: **ryškus tekstas**). Po *** simbolio einantis tekstas traktuojamas kaip sąrašo elementas, o keturiais tarpais atitraukta pastraipa arba dalis teksto įterpto į „`tekstas`“ ženklus nurodo kodo paragrafą ar dalį, kuri HTML sintaksėje patenka į `<pre>` elemento vidų. Taip pat, žymėjimo būdai gali būti apjungiami vienas su kitu, tuomet gali išgauti antraštės pavadinimą, kuris parašytas pasvirusiu bei paryškintu šriftu ir panašiai. Tai tik dalis markdown žymėjimo galimybių, kuri parodo, jog žymėjimo kalba nėra sudėtinga, bet suteikia daug galimybių.

Statinių internetinių puslapių generatorių nauda juntama ir dėl pačių statinių puslapių naudojimo. Interneto svetainės sparta išauga, nes kiekvienas puslapis egzistuoja serveryje prieš naudotojui atverčiant jį. Puslapiai yra tik nedideli tekstiniai failai, kurie pristatomi klientui per trumpą laiką. Statiniai puslapiai išvengia ir saugumo spragų sutinkamų dinaminuose internetiniuose puslapiuose, dėl puslapiuose nerandamos interakcijos su duomenų baze ar nenaudojamos dinaminės programavimo kalbas, išvengiamos potencialios saugumo spragos, kurios galėtų sukelti internetinei svetainei veikimo problemų ar kompromituoti turimus duomenis. Net jei būtų įsilaužta į internetinio puslapio serverį ir būtų ištrinti ar sugadinti visi jame esantys failai, alternatyvioje vietoje esantys puslapio failai leidžia atkurti internetinę svetainę į pradinę padėtį per trumpą laiką neprarandant jokių duomenų.

```
# An exhibit of Markdown

This note demonstrates some of what [Markdown][1] is capable of doing.

*Note: Feel free to play with this page. Unlike regular notes, this doesn't
automatically save itself.*

## Basic formatting

Paragraphs can be written like so. A paragraph is the basic block of Markdown.
A paragraph is what text will turn into when there is no reason it should
become anything else.

Paragraphs must be separated by a blank line. Basic formatting of italics and
bold is supported. This can be nested like so.

## Lists

### Ordered list

1. Item 1
2. A second item
3. Number 3
4. IV

*Note: the fourth item uses the Unicode character for [Roman numeral four][2].*

### Unordered list

* An item
* Another item
* Yet another item
* And there's more...

## Paragraph modifiers

### Code block

Code blocks are very useful for developers and other people who look at
code or other things that are written in plain text. As you can see, it uses a
fixed-width font.

You can also make inline code to add code into other things.

### Quote

> Here is a quote. What this is should be self explanatory. Quotes are
automatically indented when they are used.

## Headings

There are six levels of headings. They correspond with the six levels of HTML
headings. You've probably noticed them already in the page. Each level down
uses one more hash character.
```

An exhibit of Markdown

This note demonstrates some of what `Markdown` is capable of doing.

Note: Feel free to play with this page. Unlike regular notes, this doesn't automatically save itself.

Basic formatting

Paragraphs can be written like so. A paragraph is the basic block of Markdown. A paragraph is what text will turn into when there is no reason it should become anything else.

Paragraphs must be separated by a blank line. Basic formatting of *italics* and **bold** is supported. This *can be nested* like so.

Lists

Ordered list

1. Item 1
2. A second item
3. Number 3
4. IV

Note: the fourth item uses the Unicode character for Roman numeral four.

Unordered list

- An item
- Another item
- Yet another item
- And there's more...

Paragraph modifiers

Code block

Code blocks are very useful for developers and other people who look at code or other things that are written in plain text. As you can see, it uses a fixed-width font.

You can also make `inline code` to add code into other things.

Quote

Here is a quote. What this is should be self explanatory. Quotes are automatically indented when they are used.

Headings

There are six levels of headings. They correspond with the six levels of HTML headings. You've

3 pav. „Markdown“ naudojimo pavyzdys. Pagal: <http://www.markitdown.net/markdown>

2.3. Populiariausias „Jekyll“ generavimo įrankis

Rinkoje randama daugybė statinių puslapių generatorių, bet nenuginčijamai populiariausias iš jų – „Jekyll“ įrankis. StaticGen duomenimis, Jekyll turi daugiausiai teigiamų įvertinimų, atskirų programinio kodo atšakų GitHub sistemoje bei nedidelį kiekį klaidų lyginant su įrankio dydžiu ir populiarumu. [Sta16] Įrankis sukurtas Tom Preson-Werner ir pirmą kartą išleistas 2008 metų gale. [War08] Įrankio kūrimui panaudota Ruby programavimo kalba, turiniui kurti siūloma naudoti Markdown arba Textile anotacijas, o Liquid šablonų variklis atsakingas už tinkamo turinio įterpimą į reikiamas vietas. Šis įrankis orientuotas tinklaraščių kūrimui, turi tam optimizuotą aplankų struktūrą, kategorijas, atskirus puslapius, įrašus bei pagal poreikį kuriamus puslapio struktūros išdėstymo (angl. layout) šablonus tam, kad Jekyll sukurtos internetinės svetainės būtų kuo panašesnės į dinaminių turinio valdymo sistemų kuriamas sistemas.

```

.
├── _config.yml
├── _drafts
│   ├── begin-with-the-crazy-ideas.textile
│   └── on-simplicity-in-technology.markdown
├── _includes
│   ├── footer.html
│   └── header.html
├── _layouts
│   ├── default.html
│   └── post.html
├── _posts
│   ├── 2007-10-29-why-every-programmer-should-play-nethack.textile
│   └── 2009-04-26-barcamp-boston-4-roundup.textile
├── _data
│   └── members.yml
├── _site
├── .jekyll-metadata
└── index.html

```

4 pav. Jekyll failų struktūra, kaip pateikta: <https://jekyllrb.com/docs/structure/>

Jekyll įrankis leidžia iškviesti komandą „jekyll build“ arba „jekyll serve“ nurodytam aplankui ir jei aplankas yra tinkamas jekyll projekto aplankas, turintis reikiamą konfigūraciją ir failų struktūrą, tuomet įrankis iš duoto turinio sukuria statinę interneto svetainę „_site“ aplanke, sudarytą iš HTML puslapių ir kitų reikiamų resursų failų ir „serve“ komandos atveju startuoja lokali serverį tam, kad būtų galima naudotis ar testuoti sugeneruotą internetinį puslapį. Jekyll tikisi ketvirtame paveiksle pavaizduotos failų struktūros. Yaml faile pavadinimu „_config.yml“ galima nurodyti Jekyll konfigūracijos duomenis, daugumą galimų nustatymų galima nurodyti ir naudojantis paleidžiamaisiais parametrais, bet talpinti visą konfigūraciją vienoje vietoje yra patogiau. „_drafts“ aplanke gali būti laikomi nepaskelbti įrašai, tai tokie įrašai, kurie dar nėra baigti, jie neturi nurodytos paskelbimo datos ir jiems nebus sukuriami atskiri puslapiai. Sukūrus internetinę svetainę su „jekyll build“ komanda arba startavus serverį su „jekyll serve“ ir „--drafts“ atributu, jie bus matomi puslapyje, o sukūrimo datai bus priskirta

paskutinė failo modifikacijos data. „_includes“ kataloge laikomi komponentai, kurie gali būti įterpti į įrašus ar išdėstymus, liquid šablonų kalbos žymė „{% include failas.plt %}“ leidžia įtraukti „_includes“ aplanke esantį failą su pavadinimu „failas“ ir plėtiniu „plt“. „_layouts“ aplanke gali būti šablonai, kurie naudojami įterpiant įrašus. Išdėstymas parenkamas kiekvienam įrašui atskirai, nurodant „layout: <išdėstymo-tipas>“ YAML pirminėje medžiagoje (angl. front matter), kur <išdėstymo-tipas> yra išdėstymo pavadinimas. „_posts“ kataloge turi būti visas pagrindinis turinys – įrašai. Failai šiame aplanke privalo laikytis pavadinimo formato „metai-mėnuo-diena-pavadinimas.žymėjimo-kalba“, todėl įrašas apie naujos programavimo kalbos atsiradimą 2016 metų gegužės 24 dieną būtų pavadintas „2016-05-24-nauja-programavimo-kalba“, įrašų puslapiams generuojamas nuorodas galima konfigūruoti atskirai, bet Jekyll korektiškam funkcionavimui reikalauja laikytis nurodytos failų vardinimo tvarkos. „_data“ aplanke leidžiama talpinti tvarkingai suformatuotus duomenis naudojant yml, yaml, csv arba json plėtinius, šių failų turinys bus automatiškai įkeliamas į jekyll generatorių ir generavimo metu pasiekiamas „site.data.<failo-pavadinimas>“ kintamajame, kur „<failo-pavadinimas>“ yra aplanke randamo failo pavadinimas, pavyzdžiui „konstantos.json“ failo turinys būtų pasiekiamas „site.data.konstantos“ kintamajame. „_site“ kataloge talpinama sugeneruota internetinė svetainė, šio katalogo turinys kiekvieno generavimo metu ištrinamas ir sukuriamas naujai. „_jekyll-metadata“ faile jekyll saugo informaciją apie paskutinį kartą modifikuotus failus ir žino, kuriuos failus reikia perkurti sekančio generavimo metu. „index.html“ – tai pagrindinis failas, kuriame, naudojantis šablonais, išdėstymais ir komponentais sukuriamas pagrindinis puslapis, jei šis failas turės YAML pirminę medžiagą. Taip pat, kiekvienas failas, kuris turi YAML pirminę medžiagą, ar tuščią deklaraciją (pradžioje failo dviejose eilutėse pateikiami trys brūkšniai – „---“), bus apdorojamas jekyll generatoriaus. Jekyll generatorius apdoroja YAML pirmine medžiaga pažymėtus failus, todėl galimas „CoffeeScript“ kalba parašytų failų apdorojimas į Javascript failus ar SCSS failų – į CSS. Bet kokie kiti kataloge esantys failai generavimo metu bus kopijuojami į „_site“ katalogą.

Anot J.Dominguez, jekyll generavimo procesą sudaro keturi žingsniai, duomenų surinkimas, duomenų skaičiavimas, įrašų ir šablonų suliejimas bei rezultato generavimas. [Dom13] Pirmame žingsnyje jekyll skenuoja įrašų katalogą ir visus įrašų failus surenka kaip įrašų objektus, tuomet skenuoja ir surenka išdėstymus bei galiausiai skenuoja ir ieško atskirų puslapių. Sekančiame žingsnyje jekyll imasi duomenų apdorojimo, paima visus pagamintus

objektus, apskaičiuoja metaduomenis (nuorodas, etiketes (angl. tags), kategorijas, pavadinimus ir datas) ir sukonstruoja vieną bendrą objektą pavadintą „site“, kuris talpina visus įrašus, puslapius, išdėstymus ir atitinkamus metaduomenis. Trečiame žingsnyje jekyll pereina per visus įrašų objektus, konvertuoja juos pasitelkiant markdown arba textile sintaksę ir pradeda liquid šablonų procesą. Šablonų proceso metu jekyll inicijuoja liquid šabloną ir perduoda jam supaprastintą „site“ ir „post“ maišos (angl. hash) reprezentaciją, o prie šių objektų turima prieiga šablonuose. Paskutiniame žingsnyje vyksta šablonų pateikimas (angl. render), apdorojama šablone esanti liquid sintaksė ir išsaugoma gauta statinė puslapio reprezentacija.

3. Statinis internetinių puslapių generavimas

Internetinių puslapių rinka nepaliaujamai plečiasi, o įrenginių įvairovė, kuriais puslapiai pasiekiami didėja. Šiandien galima rasti įvairių internetinių puslapių, vieni skirti tik informacijos atvaizdavimui, kiti leidžia atlikti skaičiavimus, redaguoti vaizdo medžiagą ar bendrauti, tačiau puslapiai tampa sudėtingomis sistemomis, kurios atlieka daug skaičiavimų ne vien serveryje, bet ir naudotojo sistemoje. Matant rinkos pasikeitimus, puslapių populiarumą ir poreikį greitesnėms bei saugesnėms sistemoms buvo atsigręžta į praeitį ir prisiminti statiniai puslapiai, kurie veikia paprastai ir greitai, nereikalauja sudėtingo serverio, ir yra saugesni nei dinaminiai internetiniai puslapiai, kurių rinkoje didžioji dauguma. Buvo sugalvota alternatyva dinaminiais internetiniams puslapiams – statinių internetinių puslapių generatoriai. Tai programinė įranga, leidžianti lengviau kurti didelius internetinius puslapius pasitelkiant dinaminių internetinių puslapių generavimo principą, tik šiuo atveju puslapiai yra sugeneruojami visi iš karto ir tuomet, kaip internetinė svetainė – talpinami serveryje. Statinių puslapių generatoriai siūlo daug privalumų puslapių kūrime, talpinime ir naudojime, todėl tikslinga išsiaiškinti kokias sistemas gali pakeisti statinių puslapių generatoriais sukurtos sistemos ir kokių trūkumų toks puslapių kūrimo būdas turi. Lengviausia išsiaiškinti skirtumus ir matyti privalumus, kuomet lyginami panašūs dalykai, todėl šiame darbe dinaminis internetinis puslapis bus konvertuojamas į statinių puslapių generatoriumi sukurtą, bet pirmiau reikia išsiaiškinti, kokie yra internetinių puslapių konvertavimo žingsniai.

3.1. Dinaminio internetinio puslapio konversija į statinį

Dinaminio internetinio puslapio konvertavimas į statinį nėra trivialus ir paprastas uždavinys, priklausomai nuo esamo puslapio sudėtingumo gali tekti keisti kai kurį funkcionalumą ar atsisakyti jo dalies. Ar tinka pasirinktam puslapiui konversija į statinį galima nuspręsti tik atlikus analizę, nuo jos ir pradedama dinaminio internetinio puslapio konversija į statinį.

Dinaminio puslapio į statinį konversijos metodika

1. Pirmiausia, būtina nustatyti, ar turinys norimame perkelti puslapyje nėra priklausomas nuo dinaminio puslapio funkcionalumo, ar nėra privalomi naudotojo interakcijos su sistema būdai, kaip prisijungimas, duomenų įvedimas, sistemos funkcionalumo modifikavimas, jei šie aspektai privalomi statiniame puslapyje, reikia ieškoti alternatyvių sprendimų, kurie leistų pasiekti tokį patį ar panašų funkcionalumą, jei tokių sprendimų nėra – dinaminio puslapio paversti statiniu nepavyks.
2. Jei turinys gali būti perkeliamas į statinį puslapį, reikia suprasti, koks funkcionalumas egzistuoja puslapyje, išsiaiškinti funkcionalumą, kurio išpildyti statiniame puslapyje nepavyks, ieškoti tokiam funkcionalumui alternatyvių sprendimų. Populiariausia komentarų sistema veikianti ir statiniuose puslapiuose: „Disqus“.
3. Reikia pasirinkti statinių puslapių generatorių, kuris labiausiai tenkintų norimo perkelti dinaminio puslapio tipą, jei tai tinklaraštis, vienas populiariausių generatorių yra „Jekyll“.
4. Reikia suprasti puslapio atnaujinimo poreikį, jei reikalingas automatinis puslapio atnaujinimas atsiradus naujam turiniui, reikia statinių puslapių generavimą automatizuojančios talpinimo sistemos.
 - 4.1. GitHub Pages leidžia automatizuoti Jekyll puslapio atnaujinimą po to, kai nauji pakeitimai atsiranda saugykloje (angl. repository), kurioje talpinamas puslapis, tačiau, ši sistema turi ribotą palaikomų įskiepių kiekį. [Git16]
 - 4.2. Jei puslapis sudėtingas ir jam reikalingi įskiepiai, kurių GitHub Pages nepalaiko, tuomet „Netlify“ sistema leidžia automatizuoti statinio puslapio talpinimą po pakeitimų saugykloje.
 - 4.3. Jei automatinis atnaujinimas nėra reikalingas, galima statinį puslapį sugeneruoti lokaliai ir generavimo rezultatą patalpinti norimoje puslapių talpinimo sistemoje.
5. Reikia išsiaiškinti pasirinkto generatoriaus veikimą, konvertuoti norimo perkelti puslapio turinį į reikiamą formatą, dažniausiai tai atskiri failai aprašyti Markdown sintakse.
6. Reikia migruoti esamus dinaminio puslapio šablonus į generatoriaus šablonų sintaksę;
 - 6.1. sukurti reikiamus puslapius;
 - 6.2. sukurti išdėstymo komponentus;
 - 6.3. sukurti arba perkelti kitus resursus, reikiamus puslapio funkcionavimui, pavyzdžiui: stiliaus, JavaScript failai, paveikslėliai.

3.2. Statinio internetinio puslapio konversija į dinaminį

Ne visada statiniai puslapiai gali pakeisti dinامينius, gali prireikti konvertuoti statinį puslapį į dinaminį dėl sistemos limituoto konfigūracijos ir interakcijos lygio. Statinio internetinio puslapio konversija į dinaminį kiek paprastesnė nei atvirkščiai, nes pakeisti dinaminį funkcionalumą statiniu kartais gali būti neįmanoma arba itin sudėtinga, o statinį funkcionalumą pakeisti į dinaminį yra kur kas paprasčiau, nes galima rasti daug turinio valdymo sistemų, kurios suteikia plačias aplikacijų programavimo sąsajas (angl. API – Application Programming Interface) bei suteikia galimybę lengvai praplėsti funkcionalumą įskiepių pagalba. Kaip ir prieš dinaminį puslapių konversijos pradžią, taip ir sėkmingam statinių puslapių etapui reikalinga pirminė analizė.

Statinio puslapio konversijos į dinaminį metodika

1. Reikia pasirinkti tinkamiausią turinio valdymo sistemą (angl. CMS – Content Management System) arba kurti ją patiems. Paprastų puslapių dažniausiai pasirenkama egzistuojanti sistema, o populiariausias pasirinkimas – „Wordpress“, kuri užima apie 60% turinio valdymo sistemų rinkos. [W3t16a]
2. Reikia išsiaiškinti, ar esamo puslapio turinio konversija galima ir koku būdu bus konvertuojami duomenys. Šis žingsnis gali reikalauti daugiau laiko ir pastangų, nes neturint pakankamai statinio įrašo duomenų gali būti sunku migruoti duomenis į duomenų bazę.
3. Reikia nustatyti ar statiniame puslapyje yra naudojamas trečiųjų šalių dinaminis funkcionalumas, kaip kad komentarų sistemos, paieška ir pan. Jei minėtos sistemos egzistuoja, reikia nustatyti ar jų duomenys ir funkcionalumas yra perkeliamas į dinaminį puslapį arba ar egzistuoja alternatyvūs sprendimai, kurių funkcionalumas tenkina poreikius.
4. Jei visą reikiamą funkcionalumą galima perkelti, reikia nuspręsti, kur bus talpinamas dinaminis internetinis puslapis, nes puslapio talpinimo būdai, tinkantys statiniam puslapiui, nebūtinai tiks dinaminiam. Priklausomai nuo pasirinktos turinio valdymo sistemos reikia pasirinkti ją palaikančią talpinimo

tarnybą. Dinaminių puslapių talpinimas reikalauja serverio, gebančio apdoroti sistemai reikalingą programavimo kalbą, „Wordpress“ atveju – „PHP“ kalba bei duomenų bazės, kuri būtų palaikoma pasirinktos turinio valdymo sistemos.

5. Reikia išsiaiškinti, kaip pasirinktoje turinio valdymo sistemoje kuriami nauji šablonai. „Wordpress“ sistemoje reikia kurti naują temą, kuri gali paveldėti kitos temos konfigūraciją ir perrašyti konfigūraciją, kurią reikia;
 - 5.1. Reikia migruoti esamus statinio puslapio išdėstymo šablonus į pasirinktos sistemos šablonus, pritaikant sintaksę ar pasinaudojant sistemos suteikiamomis aplikacijų programavimo sąsajomis;
 - 5.2. Pasirinktos turinio valdymo sistemos teikiamomis priemonėmis migruoti reikiamus puslapius bei statinius resursus: paveikslėlius, stilių ar JavaScript failus. „Wordpress“ leidžia puslapius kurti pasitelkiant sistemos administracinę panelę;

3.3. Dinaminio puslapio perdarymo į statinį eksperimentas

Dinaminio puslapio konversijos į statinį puslapį eksperimentui pasirinkta tinklaraščio tipo interneto svetainė apie technologijas technews.lt naudojanti „Wordpress“ turinio valdymo sistemą. Svetainės pasirinkimą lėmė turima prieiga prie sistemos.

3.3.1. Dinaminio puslapio analizė

The screenshot shows the homepage of **technews.lt**, a Lithuanian technology news website. The main article is titled **„Be Quiet! Pure Power 9 CM“ – naujas favoritas tarp nebrangių 500 W galios maitinimo šaltinių?** (New favorite among cheap 500 W power supplies?). The article features a large image of the power supply unit and a detailed description of its features, including its 80+ Silver efficiency and modular design.

The page layout includes several key sections:

- Top Navigation:** Links for "Naujienos" (News), "Apžvalgos" (Reviews), and "Forumas" (Forum).
- Left Sidebar:** A section titled "NAUJIAUSI KOMENTARAI" (Latest Comments) showing user feedback on various articles.
- Main Content Area:** The primary article about the Be Quiet! power supply, with a large image and a detailed text description.
- Right Sidebar:** A section titled "DAUGIAU" (More) featuring a poll for "SVEIKI PHANTOM" (Welcome Phantom) and a list of "DISKUSIJOS" (Discussions) with user comments.
- Bottom Section:** A row of "NAUJIENOS" (News) items, including articles about AMD's Radeon logotipą (logo) and NVIDIA's AIDA64 duomenų bazė (database).

5 pav. technews.lt pagrindinis puslapis

Dinaminio puslapio konversija į statinį pradedama puslapio analize. 5 paveiksle pateikta technews.lt (toliau - puslapis) internetinės svetainės pagrindinio puslapio išvaizda, dėl didelio paveikslų dydžio, pavienis puslapio įrašas bei žemesnės puslapio dalis nėra įtraukiami.

1. Puslapio turinyje randamas tik statinis turinys su dažnai pasitaikančiais paveikslėliais ir kartais randamomis nuorodomis į vaizdo įrašų talpyklų sistemas, todėl puslapio turinys netrukdo vykdyti dinaminio puslapio konversijos.

2. Puslapyje randama daug dinaminio funkcionalumo reikalaujančių komponentų: paieška, naujausi komentarai, skaitomiausi įrašai, naudotojo prisijungimas, apklausa bei naujausios diskusijos, šis funkcionalumas reikalauja serveryje veikiančios programavimo kalbos bei duomenų bazės, o statinių puslapių atveju tokių galimybių nėra, todėl šio funkcionalumo reikia atsisakyti arba rasti alternatyvas.
 - 2.1. Paieškos panaudojimui statiniuose puslapiuose galima rasti įvairių „JavaScript“ kalba sukurtų įrankių, tačiau yra „lunr“² įrankis, kuris rekomenduojamas „Jekyll“ generatoriaus kūrėjų³, bet jis nėra palaikomas GitHub Pages plėtinių bibliotekos, todėl norint naudoti paiešką puslapyje su lunr įrankiu, GitHub sistema puslapių talpinimui tampa nebetinkama.
 - 2.2. Komentarų statiniuose puslapiuose leidžia pridėti „Disqus“⁴ sistema, taip pat, sistema leidžia gauti naujausius komentarus, kas leidžia realizuoti naujausių komentarų funkcionalumą puslapyje.
 - 2.3. Populiariausių įrašų funkcionalumas turi remtis dinaminiais duomenimis, vienas populiariausių variantų – Google Analytics statistika, egzistuoja įskiepis⁵, leidžiantis jekyll šablonuose turėti Google Analytics duomenis ir pateikti sąrašą populiariausių įrašų, tai leidžia realizuoti skaitomiausių įrašų funkcionalumą.
 - 2.4. Statiniai puslapiai negali turėti naudotojo prisijungimo funkcionalumo ar personalizuotos erdvės, todėl tenka atsisakyti tokių galimybių, norint konvertuoti puslapį į statinį.
 - 2.5. Apklausą veikiančią dinaminame puslapyje galima pakeisti apklausa, teikiama iš išorės, pavyzdžiui, pollmaker⁶ sistema leidžia sukurti ir teikti apklausą statiniame puslapyje.

² „Lunr“ paieškai „Jekyll“ sistemose skirtas įrankis: <https://github.com/slashdotdash/jekyll-lunr-js-search>

³ „Lunr“ paieškos sistemos naudojimo pavyzdys „Jekyll“ sistemoje <http://jekyll.tips/jekyll-casts/jekyll-search-using-lunr-js/>

⁴ „Disqus“ sistema, leidžianti statiniuose puslapiuose turėti komentarų funkcionalumą javascript pagalba: <https://disqus.com>

⁵ „Jekyll-ga“ įskiepis, leidžiantis jekyll šablonuose turėti google analytics statistiką: <https://github.com/developmentseed/jekyll-ga>

⁶ „Pollmaker“ sistema leidžia kurti apklausas ir įterpti jas į norimą puslapį: <http://www.poll-maker.com>

3. Puslapio konversijai pasirinktas „Jekyll“ generatorius dėl tinkamumo tinklaraščio tipo puslapiui, populiarumo ir daugelio randamų įskiepių bei pavyzdžių, kaip naudoti įrankį.
4. Statinio puslapio talpinimui tenka atmesti „GitHub Pages“ talpyklą dėl reikiamų įskiepių nepalaikymo, todėl pasirinkta „Netlify“ puslapio talpinimo paslauga. „GitHub“ sistema bus naudojama, kaip saugykla įrašams. Pasirinktame technologijų puslapyje dažnai kuriami nauji įrašai, todėl į „Wordpress“ sistemą įdiegtas „WordPress GitHub Sync“⁷ įskiepis, kuris kiekvieną naujai paskelbtą įrašą nusiunčia į GitHub saugyklą, todėl paliekama galimybė puslapio rašytojams kurti įrašus Wordpress sistemoje, o naudotojams juos skaityti pasirinktoje sistemoje.

3.3.2. Dinaminio puslapio konversijos eiga

Visi programinės įrangos projektai pradedami nuo lokalios aplinkos pasiruošimo, todėl sistemoje įrašyti „Ruby“ ir „Ruby-Devkit“ programinė įranga, nes Jekyll įrankiui reikalinga „Ruby“ programavimo kalbą palaikanti sistema.

Eksperimento puslapyje egzistuoja virš devynių tūkstančių įrašų. Esamų įrašų puslapyje konversijai į reikiamus „Jekyll“ įrankiui buvo bandyta naudoti „WordPress GitHub Sync“ įskiepio visų įrašų eksportavimo į „GitHub“ sistemą funkcija, tačiau dėl nepakankamų sistemos resursų (1024 MB sisteminės atminties PHP funkcijoms vykdyti), eksportavimas nebuvo įvykdytas sėkmingai. Įrašų eksportavimui buvo naudotas „Jekyll Exporter“⁸ Wordpress įskiepis, tačiau ir šiam įskiepiui pritrūko sisteminių resursų susidoroti su eksperimento puslapyje esančiu įrašų skaičiumi, todėl eksportavimas nebuvo įvykdytas sėkmingai. Esamų įrašų puslapyje konversijai į jekyll tinkančius duomenis bandytas „jekyll-import“⁹ ruby plėtinys, kuris leidžia visus įrašus eksportuoti Markdown formatu, reikalingu „Jekyll“ įrankiui. Plėtinys įdiegtas pagal jekyll¹⁰ nurodymus ir paleistas naudojantis komanda: „require 'jekyll-import'“;

⁷ „WordPress GitHub Sync“ įskiepis įrašų sinchronizavimui su „GitHub“ sistema <https://wordpress.org/plugins/wp-github-sync/>

⁸ „Jekyll Exporter“ Wordpress įskiepis įrašų eksportavimui <https://wordpress.org/plugins/jekyll-exporter/>

⁹ Jekyll-import ruby įskiepis leidžia konvertuoti Wordpress esančius įrašus į jekyll sistemai tinkamus: <https://github.com/jekyll/jekyll-import>

¹⁰ Jekyll Wordpress importo įrankio naudojimo instrukcija: <http://import.jekyllrb.com/docs/wordpress/>

JekyllImport::Importers::WordPress.run({"dbname" => "<duomenų-bazės-pavadinimas>", "user" => "<duomenų-bazės-naudotojo-vardas>", "clean_entities" => false, "comments" => false, "tags" => false, "more_excerpt" => false, "more_anchor" => false, "extension" => "md"}). Šio plėtinio pagalba buvo eksportuoti visi eksperimento puslapyje esami įrašai, nekeičiant HTML simbolių, praleidžiant komentarus, žymeles, ištraukas ir suteikiant „md“ plėtinį failams. Įrankis neturi galimybės eksportuoti specialiųjų (angl. featured) įrašų paveikslėlius. Eksportuoti įrašai buvo patalpinti „GitHub“ saugykloje ir bandyta įgalinti naujų įrašų sinchronizavimą pasitelkiant „Wordpress GitHub Sync“ įskiepi, tačiau šiuo atveju įskiepis nebesugebėjo sinchronizuoti duomenų dėl nepakankamų sistemos resursų, todėl buvo atsakyta minties sinchronizuoti naujus įrašus, sukuriamus sistemoje su konvertuotu statiniu puslapiu. Nebesant poreikio automatiniam sistemos pergeneravimui atsiradus naujam įrašui atsakyta sistemos talpinimo „Netlify“ sistemoje ir nuspręsta sugeneruotą puslapį talpinti tame pačiame serveryje, kuriame egzistuoja eksperimento sistema, dėl tiksliausių spartos vertinimo rezultatų.

Komentarams naudota „Disqus“ sistema, joje sukurta paskyra ir į ją importuoti visi komentarai iš esamo puslapio.

Perkelti visi stiliaus, „JavaScript“ failai bei paveikslėliai, reikalingi statinio puslapio funkcionavimui ir patalpinti „assets“ kataloge.

Sukurti atskiri „Jekyll“ komponentai puslapinei antraštei (head.html ir header.html), poraštei (footer.html) ir kairiajai (left-sidebar.html) šoninei juostai, kurioje patalpintas „JavaScript“ kalbos „Disqus“ komentarų įskiepis, kuris atvaizduoja naujausius komentarus sistemoje. Sukurtas komponentas dešiniajai (right-sidebar.html) šoninei juostai, kuriame patalpinta apklausa pagaminta ir talpinama „Pollmaker“ sistemoje.

Sukurti išdėstymo šablonai. Pagrindinis (default.html), kuris naudojamas visuose puslapiuose, nurodo puslapio struktūrą, įtraukia reikalingus komponentus į tinkamas vietas ir nurodo, kur puslapio struktūroje bus įterpiamas turinys. Puslapio išdėstymo šablonas (page.html), kuris nurodo, kaip atrodys puslapis (pavyzdžiui „Apie“ arba „Taisyklės“) ir kur jame įtraukiamas puslapio turinys. Įrašo šablonas (post.html), jis nurodo, kaip atrodo atvertas pavienis įrašas, kur įterpiamas įrašo turinys bei įterpia reikiamą „JavaScript“ kodą „Disqus“ sistemos komentarams.

Sukurti „Apie“ ir „Taisyklės“ puslapių statinio turinio „Markdown“ sintaksės failai su duomenimis perkeltais iš eksperimento puslapio.

Sukurtas paieškos puslapis (search.html), kuriame atvaizduojami paieškos rezultatai ir galima atlikti paiešką sistemos įrašuose. Šiame puslapyje realizuota paieška naudojantis „lunr“ paieškos įrankiu, tačiau paieška vykdoma tik įrašo pavadinime, nes pridėjus į paieškos kriterijus ir įrašų turinius vien su 375 įrašais paieška užtrunka apie septynias sekundes.

4. Eksperimento analizė

Dinaminės internetinės svetainės perkėlimo į statinę eksperimentas atliktas sėkmingai, eksperimento eigoje buvo išsiaiškinta, jog kai kurie analizės metu numatyti sprendimai netinka, neveikia arba nėra skirti susidoroti su dideliu kiekiu įrašų (virš devynių tūkstančių). Dinaminis puslapis <http://www.technews.lt/portal/> perkeltas ir patalpintas tame pačiame serveryje bei pasiekiamas adresu: <http://www.technews.lt/static/>. Statinių puslapių generatorius puslapį su 9245 atskirais įrašais generuoja apie 310 sekundžių.

Eksperimento pradžioje pastebėta, jog abu galimi „Wordpress“ sistemos įskiepai skirti duomenų eksportavimui arba sinchronizavimui su „GitHub“ saugykla „Markdown“ sintakse nesugebėjo atlikti savo darbo. Tinklaraštyje su daugiau nei devyniais tūkstančiais įrašų „Wordpress GitHub Sync“ įskiepis sėkmingai sinchronizavo naujus įrašus su saugykla, bet visų įrašų eksportas į saugyklą neveikė, klaidų įrašuose buvo parodoma klaida, kad negalima išskirti reikiamo kiekio atminties, o puslapio talpinimo serveris neteikia daugiau, nei 1GB atminties. Persikėlus duomenis į lokalų serverį ir išskyrus net 4GB atminties programinio kodo apdorojimui, klaidos įrašuose buvo randama ta pati klaida. Antras bandytas Wordpress įskiepis „Jekyll Exporter“ turėtų eksportuoti visus sistemos įrašus į „Jekyll“ generatoriui tinkamą formatą, bet bandant juo pasinaudoti buvo gaunama ta pati klaida apie nepakankamą atminties kiekį išskyrimui. Trečias eksportavimo įrankis – tai „Jekyll“ generatoriaus kūrėjų sukurtas įskiepis, naudojantis „Ruby“ kalba. Šis įrankis leido sėkmingai eksportuoti įrašus į generatoriui tinkamus failus su YAML pirmine medžiaga, kurioje pateikiama įvairi informacija apie eksportuotą įrašą, tačiau duomenyse trūksta informacijos ar nuorodos į specialųjį (angl. featured) įrašo paveikslėlį. Testavimo tikslais į keletą naujausių įrašų buvo pridėta nuoroda į

specialųjį paveikslėlį. Po sėkmingo eksporto patalpinus visus tinklaraščio įrašus „GitHub“ saugykloje ir bandant sinchronizuoti naujus įrašus, buvo pastebėta, jog gaunamos tos pačios klaidos apie negalėjimą išskirti reikiamo kiekio atminties, kaip ir eksportavimo bandytais dvejais įrankiais metu, todėl buvo atsisakyta automatinio puslapio pergeneravimo ir naujų įrašų sinchronizavimo. Atsisakius automatinio pergeneravimo poreikio buvo nuspręsta sugeneruoti puslapį ir jį talpinti tame pačiame serveryje, kur talpinamas originalus tinklaraštis dėl tikslesnių spartos rodmenų.

„Jekyll“ įrankis nesugebėjo sugeneruoti internetinės svetainės su visais įrašais, kuomet buvo bandomi sukurti kategorijų puslapiai, todėl kategorijų puslapių teko atsisakyti.

Į statinį puslapį sėkmingai perkelti visi reikalingi resursų failai, integruota komentarų sistema „Disqus“, apklausa, naudojantis „Pollmaker“ sistema.

Sukurti generatoriui reikalingi išdėstymo šablonai ir komponentai, jų sintaksė, naudojantis „Wordpress“ šablonais ir sugeneruoto puslapio kodu, adaptuota kuo tiksliau atitikti dinaminio puslapio sintaksę, elementų struktūrą, naudojamas elementų klases. Sukurta paieškos funkcija ir puslapis rezultatams bei pakartotinei paieškai pasitelkiant „JavaScript“ kalba parašytą „lunr“ įrankį. Įrankis negali atstoti dinaminio puslapio paieškos sistemos greičio aspektu, nes vos 375 įrašus turinčioje statinėje svetainėje paieška į kriterijus įtraukiant pavadinimą užtruko apie septynias sekundes, o silpnese įrenginiuose paieška gali užtrukti dar ilgiau, nes ji atliekama lankytojo naršyklėje „JavaScript“ pagalba, todėl paieškos sistemoje atsisakyta galimybės ieškoti ir turinyje. Tokie atradimai leidžia daryti išvadą, kad galinga ir greita paieška yra kur kas labiau optimizuota naudoti duomenų bazėse, o realizuojant alternatyvą statiniuose puslapiuose pasitelkiant „JavaScript“ kalbą gaunama mažiau tiksli ir daug lėtesnė paieškos sistema.

Remiantis darbo pirmo skyriaus trečio poskyrio išvadomis, aktualiausias kriterijus vertinant internetinį puslapį po konversijos yra efektyvumas, tai kriterijus, nusakantis, kiek sparčiai ir efektyviai veikia internetinis puslapis, ar greitai yra reaguojama į naudotojo veiksmus. Testavimo tikslais buvo išjungti tinklaraščio įskiepiai, kurie nebuvo realizuoti statinio puslapio atveju: skaitomiausi savaitės įrašai, prisijungimas, naujausi forumo komentarai ir populiariausios forumo temos. Nustatyti efektyvumo skirtumus tarp dinaminio tinklaraščio ir konvertuotos jo statinės reprezentacijos buvo pasirinkti trys tinklaraščio puslapiai: pagrindinis,

su dvidešimčia naujausių įrašų ir specialiųjų (angl. featured) įrašų zona su penkiais naujausiais apžvalgų kategorijos įrašais, ilgas apžvalgos tipo įrašas su daug paveikslėlių ir keletu komentarų pavadinimu „„Silicon Power S55“: šydas nuo mistinių 240 GB nutrauktas“ ir naujienos tipo įrašas, turintis nedaug teksto, vieną didelį paveikslėlį ir daug komentarų pavadinimu „Ne gamyklinio dizaino „GeForce GTX 1080“ sąrašas“. Pasirinktų puslapių užsikrovimo sparta buvo lyginama naudojantis „Chrome“ naršyklės „DevTools“ įrankio „Network“ skilties rodmenimis bei „Pingdom Website Speed Test“¹⁴ įrankiu. „Chrome“ naršyklės įrankyje svarbiausias efektyvumo kriterijus yra „DOMContentLoaded“ laikas, kuris parodo, per kiek laiko yra atsiunčiamas internetinis puslapis ir naršyklėje sukurama jo dokumento objekto modelio reprezentacija, šis rodmuo svarbiausias, nes ignoruojamas laikas sugaištas apdorojant likusius puslapio resursus, tai gali būti paveikslėliai, stilių ar JavaScript failai, kuriems užsikrauti kartu su visu puslapiu nėra privaloma, nes tie resursai gali būti nematomi naudotojui ar neaktualūs puslapio naudojimo pradžiai. „Load“ laikas, parodantis kuomet puslapis grąžina įvykį, jog jis jau yra užsikrovęs. „Finish“ laikas, parodantis, per kiek laiko buvo pilnai užkrautas puslapis. „Pingdom Website Speedtest Test“ sukonfigūruotas puslapį testuoti iš „Stockholm, Sweden“ taip sumažinant bendrą puslapio užkrovimo laiką dėl mažesnio atstumo, nei Australijoje ar Jungtinėse Amerikos Valstijose. Šio įrankio aktualiausi duomenys yra puslapio efektyvumas įvertintas skaičiumi nuo vieneto iki šimto, parodantis bendrą sistemos spartą palyginus su kitais testuotais puslapiais. Procentinė išraiška, už kiek testuotų puslapių yra greitesnis testuojamasis puslapis. Procentinė išraiška laiko, praleisto laukiant atsakymo iš serverio. Visi testai kartojami tris kartus ir imama vidutinė reikšmė, kiekvienas puslapio perkrovimas vykdomas ištuštinant podėlį (angl. cache) ir atliekant pilną perkrovimą (angl. hard reload). Rezultatai pateikiami 2 lentelėje, kur puslapis nr.1 – pagrindinis tinklaraščio puslapis, puslapis nr.2 - didelis įrašas su daug nuotraukų pavadinimu „„Silicon Power S55“: šydas nuo mistinių 240 GB nutrauktas“, puslapis nr.3 – įrašas su mažai teksto, didele nuotrauka ir daug komentarų pavadinimu „Ne gamyklinio dizaino „GeForce GTX 1080“ sąrašas“. „PWST“ trumpinys lentelėje - „Pingdom Website Speedtest Test“. Testavimas atliktas kompiuteriu, turinčiu Intel i5-4690K procesorių dirbantį 4,5Ghz dažniu, 8GB RAM atminties ir naudojančią Windows 10 64bit sistemą.

¹⁴ „Pingdom Website Speed Test“ įrankis matuojantis internetinių puslapių spartą: <http://tools.pingdom.com/fpt/>

Efektyvumo analizės rezultatai

2 lentelė: Dinamini ir statinių internetinių puslapių spartos palyginimas

	Puslapis nr.1		Puslapis nr.2		Puslapis nr.3	
	Dinaminis	Statinis	Dinaminis	Statinis	Dinaminis	Statinis
„DOMContentLoaded“ laikas, s	1,410	0,391	0,873	0,402	0,850	0,358
„Load“ laikas, s	1,583	0,555	1,277	0,600	1,253	0,553
„Finish“ laikas, s	1,607	0,572	1,301	1,363	1,284	1,230
„PWST“ efektyvumas	85	83	85	86	84	73,7
„PWST“ greičiau nei % puslapių	78,7	88	74	81,3	80,7	94
„PWST“ serverio laukimo laikas, %	54,36	30,97	58,6	30,97	55,30	46,14

Naudojantis 2 lentelėje pateikiamais internetinių puslapių analizės rezultatais galima teigti, jog statinis puslapis visų testuotų puslapių atvejais greičiausiai užkrauna ir apdoroja dokumento objekto modelį bei pateikia puslapio užsikrovimo pabaigos įvykį, tai įvyksta apie du ar net tris kartus greičiau, tačiau tik pagrindinio tinklaraščio puslapio atveju statinis puslapis stipriai aplenkė dinaminį puslapį pilno puslapio užkrovimo trukme. Tokia tendencija rodo, kad tinklaraštyje naudojami Wordpress komentarai veikia greičiau, nei išorinė „Disqus“ komentarų, nes jos krovimasis užtęsia „Finish“ puslapio laiką antro ir trečio puslapio atvejais, tačiau puslapis yra naudojamas ir nesulaukus, kol komentarai bus užkrauti ir parodyti. „Pingdom Website Speedtest Test“ testo efektyvumo rezultatas pirmųjų dviejų puslapių atveju yra beveik lygus, tačiau trečio puslapio atveju rezultatas dešimčia taškų mažesnis, tai lėmė faktas, jog puslapis yra nedidelis, su mažai resursų, tačiau atskirų JavaScript failų daug ir įrankis siūlo failus apjungti į vieną, kur galima. Puslapių testavimas su „Pingdom Website Speedtest Test“ įrankiu dar kartą parodo, kad statiniai puslapiai buvo atsiųsti kur kas greičiau, nei dinaminiai, nes įrankis vertina puslapio užsikrovimo spartą iki kol puslapis išsiunčia „load“ įvykį, rodantį, kad jis užsikrovė. Serverio laukimo laiko kriterijus visuomet buvo statinių puslapių naudai, tai procentas laiko, kurį buvo laukiama atsakymo iš serverio, nuo bendro

puslapio užkrovimo laiko. Nuo 9 iki 28 procentų besiskiriančios reikšmės parodo, kad statiniai puslapiai kur kas mažiau apkrauna serverius duomenų apdorojimu.

Dinaminių ir statinių internetinių puslapių efektyvumo analizė atskleidė, kad statiniai puslapiai tikrai turi privalumų prieš dinامينius puslapius. Statinių puslapių atveju buvo kur kas greičiau apdorojamas puslapio dokumento objekto modelis, taip pat, daug greičiau išsiunčiamas puslapio užsikrovimo pabaigos įvykis, tik puslapių su „Disqus“ komentarų sistema pilno užsikrovimo pabaigos laikas susilygino su dinaminų puslapių. Statiniai puslapiai ne tik užsikrauna keletą kartų greičiau, bet ir reikalauja, ir sunaudoja mažiau serverio resursų už dinامينius puslapius. Statiniai puslapiai puikiai tiktų sistemoms, kurios nereikalauja daug dinaminio funkcionalumo, kaip kad dokumentacijų ar žinių bazės puslapiai, portfelio (angl. portfolio) ar kiti nedideli puslapiai.

Analizė parodė, jog statinių puslapių generavimas nebūtų tinkamas pasirinkimas didelėms informacinėms sistemoms, elektroninėms parduotuvėms ar kitiems puslapiams su daug įrašų. Su dideliu kiekiu turinio būtų sudėtinga susidoroti paieškai, taip pat, problemų kėlė ir bandymas sukurti kategorijų puslapius, nes įrašų tiek daug, jog sistemos leidžiamos išskirti atminties kiekis generatoriui nebuvo pakankamas.

Rezultatai ir išvados

Rezultatai:

1. Apibrėžti statiniai ir dinaminiai puslapiai (1.1 – 1.2 skyriai);
2. Apibrėžti kokybiško internetinio puslapio kriterijai (1.3 skyrius);
3. Statiniai ir dinaminiai internetiniai puslapiai palyginti pagal kokybės kriterijus (1.3 skyrius);
4. Išskirti statinių ir dinaminių internetinių puslapių privalumai ir trūkumai (1.4 skyrius);
5. Apibrėžti statinio ir dinaminio puslapio gavimo (žingsnių seka, kuri įvykdoma, kol naudotojas pamato internetinį puslapį) keliai (1.4 skyrius);
6. Suprastas statinių internetinių puslapių generavimo principas ir nauda (2.1 – 2.2 skyriai);
7. Išanalizuoti populiariausio statinių puslapių generatoriaus veikimo principai (2.3 skyrius);
8. Apibrėžtos statinio ir dinaminio internetinio puslapio konversijos metodikos (3.1 – 3.2 skyriai);
9. Atliktas dinaminio internetinio puslapio konversijos į statinį internetinį puslapį eksperimentas (visas 3.3 skyrius);
10. Atlikus konversijos eksperimentą rasti statinių puslapių generavimo privalumai ir trūkumai prieš dinامينius puslapius. (3.3.2 ir 4 skyriai)

Išvados:

1. Internetinių puslapių populiarumas ir kiekis auga, o jų naudojimo įpročiai keičiasi, pagrindine internetinių puslapių peržiūros terpe tapo mobilieji įrenginiai;
2. Duomenų cirkuliacijos internete mastai auga kiekvienus metus, o jiems apdoroti reikalingos vis galingesnės sistemos;
3. Internetinių puslapių kaip sistemos populiarumas auga, o tokie puslapiai dažniausiai reikalauja sudėtingos dinaminės sistemos serveryje, kas atveria daug galimybių, bet ir sukelia galimų problemų rizika. Internetiniuose puslapiuose galima atlikti daugybę

darbų ar rasti įvairių pramogų, bet juose gali tykoti ir saugumo pavojai, nes jokia programinė įranga nėra pilnai saugi;

4. Internetinių puslapių panaudojimo ir galimybių pasikeitimai nuo jų atsiradimo 1989 metais iki dabar pasikeitė drastiškai. Dabar populiariausi, dinaminiai, puslapiai leidžia išpildyti bet kokią dinaminį funkcionalumą puslapyje, kurį įmanoma išpildyti pasitelkiant programavimo kalbas;
5. Kuomet egzistuoja tiek daug internetinių puslapių (daugiau nei 250 milijonų), nenuostabu, jog nevisi jie yra kokybiški. Kokybiški internetiniai puslapiai gali būti vertinami pagal aktualumo, naudingumo, patikimumo, specializacijos, architektūros, navigacijos aiškumo, efektyvumo, išdėstymo (angl. layout) bei animacijos kriterijus, bet internetinių puslapių konversijos tarp dinaminio ir statinio atveju tik efektyvumo kriterijus yra aktualus, nes likusieji kriterijai gali būti nepakitę, jei puslapis konvertuojamas išbaigta;
6. Dinaminių ir statinių internetinių puslapių gavimo kelias atskleidžia skirtumus, kurie turi įtakos puslapių spartai ir saugumui. Dinaminiai puslapiai dažnai turi didesnį persiunčiamų duomenų kiekį, duomenys ir užklausos yra apdorojamos serveryje programinės įrangos bei vykdomos transakcijos su duomenų baze. Visi žingsniai dinaminių puslapių žingsniai padidina saugumo spragų galimybę bei sulėtina puslapio gavimą;
7. Nevisi puslapiai turi naudos iš dinaminių puslapių funkcionalumo, todėl pastebimas statinių puslapių naudojimo populiarėjimas, jų kūrimui pasitelkiant statinių internetinių puslapių generatorius. Generatoriai palengvina internetinių puslapių kūrimo etapą pasitelkdami dinaminių puslapių ideologijas: naudojami išdėstymo šablonai, komponentai, turinys atskiriamas nuo realizacijos, bet didžiausias skirtumas nuo dinaminių puslapių toks, jog generatoriai sukuria visos internetinės svetainės puslapius generavimo metu ir jie yra visuomet prieinami naudotojui;
8. Statinių internetinių puslapių generatoriai ne tik palengvina puslapių kūrimo procesą, bet leidžia kurti turinį ir techninių žinių neturintiems žmonėms, pasitelkiant paprastą duomenų failų sintaksę (pavyzdžiui Markdown), kurią lengva perprasti ir įvaldyti, o tokie įrankiai, kaip „Prose“ leidžia lengvai keisti turinį, esantį „GitHub“ saugykloje, o automatinio regeneravimo būdu pakeitimai gali būti atvaizduojami sugeneruotame statiniame puslapyje;

9. „Jekyll“ statinių puslapių generavimo įrankis yra pats populiariausias ir orientuotas į tinklaraščių kūrimą. Juo paremtas „GitHub Pages“ sistemos veikimas;
10. Dinaminio puslapio konversija į statinį ar statinio konversija į dinaminį turi būti pradedama nuo analizės. Tik išsiaiškinus sistemos poreikius, privalomą konvertuoti funkcionalumą ir radus įrankius bei įskiepius, pakeisiančius dinaminį funkcionalumą galima teisingai atlikti puslapių konversiją;
11. Dinaminių puslapių duomenų eksportavimo ar sinchronizavimo įrankiai yra sunkiai konfigūruojami ir modifikuojami bei sunkiai susidoroja su dideliu kiekiu įrašų (virš devynių tūkstančių eksperimento metu);
12. Daugumai dinaminio funkcionalumo puslapiuose galima rasti alternatyvų statiniams puslapiams, komentarai, paieška sistemoje, balsavimas ir panašiai. Tačiau visas šis funkcionalumas įgalinamas „JavaScript“ kalbos pagalba, todėl puslapio veikimo sparta pradeda labiau priklausyti nuo sistemos, kurioje naršomas puslapis;
13. Dinaminis puslapis konvertuotas į statinį pasižymi keletą kartų didesne puslapio užkrovimo sparta, ypatingai tuomet, kai puslapyje reikia atvaizduoti daug įvairios informacijos esančios duomenų bazėje. Konvertuotas statinis puslapis juntamai mažiau apkrauna serverį.
14. Statinio puslapio dinaminis funkcionalumas, pavyzdžiui, komentarai, naudojant išorinę sistemą gali bendro puslapio užsikrovimo spartos persvarą prieš dinaminį sumažinti iki minimumo, bet asinchroniniu būdu užkraunamas funkcionalumas netrukdo puslapiu naudotis greičiau, nei jis yra pilnai pakraunamas;
15. Statinių puslapių generavimas gali būti puiki alternatyva tinklaraščio, portfelio (angl. portfolio) tipo puslapiams, dokumentacijos ar žinių bazės puslapiams. Tokie puslapiai, kuriems nereikalingas platus dinaminis funkcionalumas, įgautų papildomos spartos, saugumo, dėl sistemos paprastumo bei liktų lengvai atnaujinami;
16. Statinių puslapių generavimas nėra gera alternatyva sudėtingoms informacinėms sistemoms, elektroninėms parduotuvėms, puslapiams, teikiantiems interaktyvų funkcionalumą ar kitoms sistemoms, kuriose reikalingas platus dinaminio funkcionalumo spektras ir panaudojimas;
17. Statiniai puslapiai su plačiomis dinaminio funkcionalumo alternatyvomis gali būti blogas pasirinkimas mobiliuose platformose dėl silpnesnės skaičiuojamosios galios, kurios gali pritrūkti apdorojant didelius kiekius „JavaScript“ kodo.

Literatūros sąrašas

- [All16] R. Allen. What happens online in 60 seconds?, <http://www.smartinsights.com/internet-marketing-statistics/happens-online-60-seconds/>. 6,2MB, 2016-04-20
- [Bou15] E. Bouças. An Introduction to Static Site Generators, <https://davidwalsh.name/introduction-static-site-generators>. 1,0MB, 2015-05-20
- [Cer16] CERN. The birth of the web, <http://home.cern/topics/birth-web>. 1,6MB, 2016-05-30
- [Cha16] D. Chaffey. Mobile Marketing Statistics compilation, <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>. 5,5MB, 2016-04-27
- [Dom13] J. Dominguez. How jekyll works, <http://jekyllbootstrap.com/lessons/jekyll-introduction.html>. 332KB, 2013
- [Ewe13] T. Ewer. Is WordPress Secure?, <https://managewp.com/is-wordpress-secure>. 1,2MB, 2013-05-13
- [Git16] GitHub Pages. Dependency versions, <https://pages.github.com/versions/>. 54KB, 2016-05-30
- [Goo16] Google Developers. Mobile Friendly Websites, <https://developers.google.com/webmasters/mobile-sites/>. 1,3MB, 2016-05-09
- [Int16] Internet live stats. Total number of Websites, <http://www.internetlivestats.com/total-number-of-websites>. 1,9MB, 2016-05-09
- [MLD04] V. S. Moustakis, C. Litos, A. Dalivigas, L. Tsironis. Website quality assesment criteria. Technical University of Crete, 2004, Greece
- [Pin07] Pingdom blog. A history of the dynamic web, <http://royal.pingdom.com/2007/12/07/a-history-of-the-dynamic-web/>. 1,6MB, 2007-12-07

- [Sta16] StaticGen. Top open-source static site generators, <https://www.staticgen.com>. 883KB, 2016-05-24
- [W3tT16] W3Techs. Usage of server-side programming language for websites, http://w3techs.com/technologies/overview/programming_language/all. 45KB, 2016-05-11
- [W3tT16a] W3Techs. Market share trends for content management systems for websites, http://w3techs.com/technologies/history_overview/content_management. 119KB, 2016-05-11
- [War08] T. P. Warner. Blogging Like a Hacker, <http://tom.preston-werner.com/2008/11/17/blogging-like-a-hacker.html>. 216KB, 2008-11-17
- [Wik16] Wikipedia. GoDaddy, <https://en.wikipedia.org/wiki/GoDaddy>. 665KB, 2016-05-2809
- [Wik16a] Wikipedia. Common Gateway Interface, https://en.wikipedia.org/wiki/Common_Gateway_Interface. 396KB, 2016-04-06
- [Wik16b] Wikipedia. Denial-of-service attack, https://en.wikipedia.org/wiki/Denial-of-service_attack. 530KB, 2016-05-30