VILNIAUS UNIVERSITETAS MATEMATIKOS IR INFORMATIKOS FAKULTETAS PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

"Blockly" taikymo galimybės turinio valdymo sistemų duomenų atvaizdavimui

"Blockly" Application to Display Content Management Systems Data

Bakalauro baigiamasis darbas

Atliko:	4 kurso 1 grupės studentė	
	Neringa Geigalaitė	(parašas)
Darbo vadovas:	Lekt. Irus Grinis	(parašas)
Darbo recenzentas:	Doc. Dr. Kristina Lapin	(parašas)

Santrauka

Šis darbas apžvelgia "Blockly" įrankio, skirto kurti pritaikytas blokais paremtas vaizdines kalbas, taikymo galimybes turinio valdymo sistemų duomenų atvaizdavimui generuoti. Darbe pasirinkta kurti temas "WordPress" turinio valdymo sistemai ir generuoti internetines svetaines "Joomla", "Drupal" ir "WordPress" API sąsajos suteiktiems duomenims. Šio darbo tikslas yra išanalizuoti ar "Blockly" yra pritaikomas pasirinktoje srityje. Tikslui pasiekti yra nagrinėjami "Blockly" elementų kūrimo principai, analizuojami egzistuojantys "WordPress" temų generavimo įrankiai ir temų kūrimo principai. Norint suprasti įrankio pritaikymo galimybes, yra sukurtos dvi sistemos, kurios apima tradicinės ir *headless* turinio valdymo sistemos duomenų atvaizdavimo lygmenį. Darbe yra aprašyti "Blockly" vaizdinės programavimo kalbos kūrimo principai ir su kokiais ribojimais galima susidurti taikant "Blockly" šioje srityje.

Raktiniai žodžiai: Blockly, vaizdinė programavimo kalba, turinio valdymo sistema.

Summary

This paper reviews the application of the Blockly tool to generate a data representation layer for content management systems. In the paper, themes are created for the WordPress content management system and websites are generated for the data provided by the Joomla, Drupal, and WordPress APIs. The aim of this work is to analyze whether Blockly is applicable in the specified field. To achieve this goal, this paper analyzes the principles of creating Blockly elements, overviews the existing WordPress theme generator tools and its theme creation principles. This paper covers two created systems. The first one generates themes for a traditional content management system. The other one generates a representation layer for several headless CMS. The paper describes the principles of developing a visual programming language and observes the limitations that can be encountered while applying Blockly in the specified area.

Key words: Blockly, visual programming language, content management system (CMS)

TURINYS

ĮVA	ADAS Problema ir jos aktualumas Tikslas ir uždaviniai	4 4 4
1.	"BLOCKLY" ĮRANKIS 1.1. Blokų kūrimas 1.2. Programinio kodo generavimas	6 6 8
2.	TURINIO VALDYMO SISTEMOS	10
3.	 "BLOCKLY" PRITAIKYMAS TVS TEMŲ GENERAVIMUI 3.1. "WordPress" temos kūrimas 3.1.1 Temos struktūra 3.1.2 Temos funkcionavimas 3.2 Temų generavimo įrankių kriterijai 3.3 Egzistuojantys temų generavimo įrankiai 3.3.1 "Divi" 3.3.2 "Elementor Pro" 3.3.3 "Thrive Theme Builder" 3.3.4 Egzistuojančių temų generavimo įrankių palyginimas 3.4. "WordPress" temų generavimui pritaikytas "Blockly" įrankis 3.4.1 Temų generavimui pritaikyta vaizdinė kalba 3.4.2 Naudotojo sąsaja 3.5 Gautų rezultatų apibendrinimas 	11 11 13 13 14 14 15 16 16 17 17 22 23
4.	 "BLOCKLY" PRITAIKYMAS TVS DUOMENŲ ATVAIZDAVIMUI KURTI 4.1. Programinė sąsaja 4.2. Sistemos struktūra 4.3. "WordPress", "Joomla" ir "Drupal" duomenų atvaizdavimui kurti pritaikytas "Blockly" 4.4. Gautų rezultatų apibendrinimas 	25 25 26 27 29
RE	ZULTATAI IR IŠVADOS	30
ŠA	LTINIAI	31
PR	IEDAI	33

Įvadas

Problema ir jos aktualumas

Blokais paremtas vaizdinis programavimas tapo susipažinimo su programavimu *lingua franca*. Yra ištirta, kad patirtis su blokais paremtomis programavimo kalbomis padeda greičiau įsisavinti ir išmokti tekstu paremtas kalbas. Tačiau vaizdinis programavimas yra naudojamas ne tik edukaciniais tikslais. Ši programavimo rūšis apima sritis, kuriose plačiai naudojamos piktogramos arba diagramos informacijai perduoti, bei suteikiama sąsaja komunikacijai tarp žmogaus ir kompiuterio. Vaizdinis programavimas yra naudojamas tokiose srityse kaip grafinės naudotojo sąsajos generavimas, duomenų bazės manipuliacija ir kita.

Vaizdinės programavimo kalbos kūrimui galima naudoti įrankius, kurie suteikia grafiniais elementais paremtą aplinką ir leidžia tinkinti egzistuojančius arba kurti naujus elementus. Vienas iš tokių įrankių yra "Blockly" – naudotojo sąsaja, skirta kurti blokais paremtas vaizdines kalbas, kurių elementai generuoja programinį kodą [BGK⁺17; Blo20; FTV02].

"Blockly" yra naudojamas įvairiose srityse: daiktų interneto programų kūrimui, darbo įrankių integracijų automatizavimui, programavimo mokymui, mobiliųjų programėlių kūrimui ir kita. Kadangi vaizdinės programavimo kalbos yra naudojamos sąsajos su žmogumi generavimui, "Blockly" taip pat galėtų būti pritaikytas kurti blokais paremtoms vaizdinėms programavimo kalboms, kurios generuotų internetinių puslapių naudotojo sąsajas [Blo20].

Internetinėms svetainėms kurti dažnai naudojamos turinio valdymo sistemos (TVS). Tai yra programinė įranga sukurta sekti ir valdyti internetinių puslapių turinį be techninių puslapių kūrimo žinių. Dėl skaitmeninių rinkodaros sprendimų poreikio, augančio skaitmeninio pardavimų sektoriaus ir naudotojų poreikio daugiakanalėms (angl. *omnichannel*) patirtims, turinio valdymo sistemų naudojimas auga. Iki 2026 metų šios rinkos planuojamas augimas yra 17 procentų. Kadangi šių sistemų populiarumas auga, atsiranda poreikis priemonėms, su kuriomis būtų galima atvaizduoti jose sukurtus duomenis [Sur20].

Taigi, "Blockly" galėtų būti pritaikytas turinio valdymo sistemų duomenų atvaizdavimui, kaip priemonė kurti naudotojo sąsajos elementus.

Tikslas ir uždaviniai

Šio darbo tikslas – išanalizuoti "Blockly" įrankio pritaikymo galimybes turinio valdymo sistemų duomenų atvaizdavimui kurti.

Tikslui pasiekti yra atliekami du bandymai: "Blockly" pritaikymas pasirinktos turinio valdymo sistemos temų generavimui ir "Blockly" pritaikymas turinio valdymo sistemų suteiktos sąsajos duomenims atvaizduoti.

Darbo uždaviniai:

- 1. Išnagrinėti "Blockly" įrankį.
- 2. Išanalizuoti egzistuojančius temų generavimo įrankius.

- 3. Išnagrinėti, kaip kuriamos pasirinktos turinio valdymo sistemos temos.
- 4. Sukurti pasirinktos TVS temų generavimo įrankį ir įvertinti tokio įrankio galimybes pagal atliktą egzistuojančių TVS analizę.
- 5. Pritaikyti "Blockly", kaip priemonę TVS programinės sąsajos suteiktų duomenų atvaizdavimui kurti.

1. "Blockly" įrankis

"Blockly" yra "Google" sukurta atvirojo kodo biblioteka, kuri skirta į programą pridėti blokais paremtą programavimo aplinką. Ši biblioteka suteikia blokų redagavimo naudotojo sąsają ir kodo generavimui į tekstu paremtas kalbas skirtą karkasą. Kadangi "Blockly" parašyta "JavaScript" programavimo kalba, ją galima integruoti į bet kokį internetinį puslapį [PFM17].

"Blockly" nėra nei kalba, nei sistema, kuri jau būtų paruošta galutiniam naudotojui. Ji suteikia gramatiką ir programavimo atvaizdavimą, kurį programuotojai gali naudoti savo sistemose. Programinis kodas yra atvaizduojamas blokais, kurie gali būti nutempti į skirtingas ekrano vietas. Blokai turi susijungimo taškus, kuriais jie yra jungiami su kitais blokais. Norint naudoti "Blockly", programuotojai turi šiam įrankiui sukurti žodyną ir nuspręsti, kaip sugeneruotas kodas bus vykdomas [PFM17].

1.1. Blokų kūrimas

Blokus galima kurti naudojantis "Blockly Developer Tools" įrankiu arba rašant kodą ranka pagal "Blockly" suteiktą blokų kūrimo API.

Naudojant API, blokai aprašomi naudojant JSON objektus arba JavaScript funkcijas. JSON formatas yra prioritetinis metodas grafiniams objektams aprašyti, tačiau jis neleidžia tiesiogiai aprašyti pažengusių funkcijų (*mutators* ir *validators*). Programos, naudojančios JavaScript blokų kūrimui, gali juos aprašyti tiesiogiai "Blockly" API žemesniajame lygyje. Pasirinkus blokus aprašyti JSON formatu ir prireikus įgyvendinti pažengusias funkcijas, jas reikės perrašyti JavaScript kalba [Blo20].

"Blockly Developer Tools" yra žiniatinklio pagrindu (angl. *web-based*) sukurtas įrankis, kuris automatizuoja blokų konfigūravimo procesą: blokų ir įrankių dėžutės kūrimą, darbo srities tinkinimą. "Blockly" programavimo procesas, naudojantis šiuo įrankiu, sudarytas iš trijų žingsnių [Blo20]:

- sukurti bloką;
- sukurti įrankių dėžutę ir numatytąją darbo sritį;
- sukonfigūruoti darbo sritį;

"Blockly Developer Tools" įrankiu yra kuriami nauji blokai "Block Factory" skiltyje. Vienoje pusėje yra darbo sritis, kurioje iš blokų yra jungiamas naujas grafinis elementas (1 pav.). Čia galima nurodyti jo pavadinimą, kokias įvestis jis priima, parašyti naudojimo patarimą (angl. *tooltip*), nurodyti pagalbinę nuorodą naudotojui, bei nustatyti spalvą.



1 pav. "Block Factory" darbo sritis

Kitoje pusėje yra matoma, kaip atrodo naujas blokas (2 pav. 1 blokas), jo aprašymo kodas su JavaScript arba JSON (2 pav. 2 blokas) ir jo generuojamas kodas, kuris gali būti eksportuotas į 5 kalbas (2 pav. 3 blokas).



2 pav. "Block Factory" sukurto bloko rezultatas

1.2. Programinio kodo generavimas

Kiekvienas blokas turi jam atitinkantį kodo fragmentą. Jungiant blokus, yra sujungiamas ir jų kodas. Kartu šie kodo fragmentai gali sudaryti vykdymui paruoštą programinį kodą, kuris sugeneruojamas JavaScript, Python, PHP, Lua ir Dart kalbomis [Blo20].

Bet kurio bloko kodo generavimo pirmas darbas yra surinkti visus argumentus ir laukelių duomenis. Yra trys funkcijos šiam darbui atlikti (1 lentelė):

Metodas	Paskirtis	
getFieldValue	Šis metodas grąžina nurodyto laukelio reikšmę.	
valueToCode	Šis metodas suranda bloką, kuris yra prijungtas prie vie-	
	nos reikšmės įvesties eilutės nurodytu pavadinimu, suge-	
	neruoja kodą tam blokui ir grąžina kodą kaip tekstinę ei-	
	lutę. Jeigu blokas nėra prijungtas, metodas grąžina nulinę	
	reikšmę.	
statementToCode	Šis metodas suranda visus įdėtinius blokus, kurie yra pri-	
	jungti prie nurodytos kelių reikšmių įvesties eilutės. Tada	
	sugeneruoja tų blokų kodą ir grąžina jį suformatuotą kaip	
	tekstinę eilutę. Jeigu nėra įdėtinių blokų, metodas grąžina	
	tuščią eilutę.	

1 lentelė. Metodai, skirti bloko reikšmes išgauti. [Blo20]

Kai visi argumentai yra surenkami, jie yra naudojami galutiniam kodui sujungti. Programuotojas kontroliuoja, kaip tie argumentai yra panaudojami, tačiau pats blokų kodo jungimas yra automatizuotas bibliotekos vykdomas procesas. Galutinis kodas yra visų sujungtų blokų programinio kodo vienetas [Blo20].

2. Turinio valdymo sistemos

Turinio valdymo sistema yra programinė įranga sukurta sekti ir valdyti internetinių puslapių turinį be techninių puslapių kūrimo žinių. Turinys gali būti tekstas, nuotraukos, garso ir vaizdo įrašai, dokumentai ar kita. Internetinio puslapio vizualinis pateikimas, išdėstymas ir struktūra pakeičiama automatiškai pagal puslapio naudojamą temą, kuri atskiria turinį nuo puslapio atvaizdavimo. TVS galima išskirti į tradicines ir *headless* [MJS14].

Tradicinės TVS yra monolitas, kuris sujungia išorinę (angl. *front end*) ir vidinę (angl. *back end*) svetainės puses vienoje kodo bazėje. Ši kodo bazė talpina viską nuo duomenų bazės iki atvaizdavimo lygmens. Tokia TVS svetainė yra patiekiama vienoje sistemoje ir naudotojo sąsaja yra kontroliuojama per temas ir šablonus. Šioms svetainėms kurti reikalingos konkrečios TVS temos kūrimo žinios ir tas pats turinys įprastai negali būti patiekiamas kituose įrenginiuose, nei kuriama svetainė [Mel; UDi].

Headless TVS išsiskiria tuo, kad ji nėra susieta su išorine svetainės puse. Tradicinė TVS bendrai valdo kuriamą turinį ir išorinę pusę, o *headless* TVS valdo tik kuriamą turinį. Kai tokioje sistemoje yra sukuriami duomenys, jie yra paskelbiami per sistemos programinę sąsają, ko rezultate yra leidžiama pasirinkti, kuriems atvaizdavimo pabaigos taškams jie yra siunčiami [Mel; UDi].

Tradicinių TVS duomenų atvaizdavimas yra konfigūruojamas su šablonais arba temomis, o *headless* TVS duomenys atvaizduojami kuriant atskirą naudotojo sąsają, kurioje jie yra surenkami ir atvaizduojami. Ta pati turinio valdymo sistema gali būti naudojama ir kaip *headless*, ir kaip tradicinė TVS. Šiame darbe yra siekiama pritaikyti vaizdinę programavimo kalbą abiejų TVS tipų duomenų konfigūravimui internetinėje svetainėje (3 pav.):

- duomenų atvaizdavimui, kuriant tradicinės TVS temas;
- duomenų atvaizdavimui, kuriant headless TVS svetaines.



3 pav. "Blockly" taikymas TVS duomenų atvaizdavimui kurti [Mel]

3. "Blockly" pritaikymas TVS temų generavimui

Pirmasis su "Blockly" vykdomas bandymas yra šio įrankio pritaikymas pasirinktos TVS temoms generuoti. Šiame darbe pasirinkta šablonus kurti labiausiai naudojamai atvirojo kodo turinio valdymo sistemai "Wordpress" [Sur20].

Norint pritaikyti "Blockly" "Wordpress" temų generavimui, reikia išsiaiškinti tokio įrankio kriterijus ir išnagrinėti, kokias galimybes siūlo jau egzistuojančios temų generavimo priemonės. Remiantis šia analize, galima stebėti, kokie yra "Blockly" ribojimai ir privalumai. Šio bandymo tikslas nėra atkartoti jau egzistuojančias priemones, bet suprasti kokios yra "Blockly" galimybės šioje srityje.

3.1. "WordPress" temos kūrimas

Norint sukurti temų generavimui pritaikytą įrankį reikia išanalizuoti "WordPress" temų kūrimo principus.

"WordPress" temos yra failai, kurie kartu jungiami sudaro "WordPress" internetinio puslapio dizainą ir funkcionalumą. Temos sukuria unikalų internetinio puslapio vaizdavimą, kurį gali pritaikyti ir programavimo žinių neturintys sistemos internetinės svetainės savininkai. Jos keičia internetinio puslapio atvaizdavimą nedarant įtakos egzistuojančiai programinei įrangai [Wor20].

Šiame poskyryje aprašomi "WordPress" temų kūrimo pagrindai.

3.1.1. Temos struktūra

Temos standartiškai sudaromos iš trijų tipų failų [Wor20]:

- stilių lentelių, kurios valdo svetainės dizainą ir komponentų išdėstymą;
- šabloninių failų, kurie kontroliuoja kaip svetainės atvaizduoja informaciją iš "WordPress" duomenų bazės;
- temos funkcijų failų.

Šabloniniai failai yra moduliniai, daugkartinio naudojimo failai, skirti generuoti tinklalapius "WordPress" svetainėse. "WordPress" naudoja užklausų eilutes nuspręsti, kurie šablonai ar šablonų rinkiniai turi būti naudojami atvaizduoti puslapiui. Užklausų eilutė yra informacija, esanti nuorodoje į kiekvieną svetainės dalį. "WordPress" ieško žemyn per šablonų hierarchiją, kol suranda atitinkamą šabloninį failą [Worc].

Norint nustatyti, kurį šabloninį failą naudoti, yra vykdomi šie veiksmai [Worc]:

- kiekviena užklausos eilutė palyginama su užklausos tipu tam, kad būtų nuspręsta, kurio puslapio yra prašoma;
- pasirenkamas šablonas, priklausomai nuo eilės, kuri yra nurodyta šablonų hierarchijoje;
- ieškoma šabloninių failų su nurodytais pavadinimais tos temos kataloge ir panaudojamas pirmas tinkantis rastas šabloninis failas, kaip nurodyta hierarchijoje.

Yra išskiriami pagrindiniai šablonai, kurie padengia esminį temos funkcionalumą. Pirminių šabloninių failų paskirtis:

Šablonas	Paskirtis	
index.php	Pagrindinis šabloninis failas. Naudojamas visoms užklausoms, jei nėra randa-	
	mas konkretesnis šablonas.	
archive.php	Archyvo šablonas yra naudojamas tada, kai svetainės lankytojas užklausia įrašų	
	pagal kategoriją, autorių arba datą.	
single.php	Vieno įrašo šablonas, kai svetainės lankytojas užklausia peržiūrėti vieną įrašą.	
page.php	Puslapio šablonas, kuris yra naudojamas kai lankytojas užklausia individualių	
	puslapių, kurie yra įtaisyti (angl. built-in) šablonai.	
home.php	Pradžios tinklalapio šablonas, kuris standartiškai yra pradinis puslapis. Jeigu	
	"WordPress" nėra nustatyta naudoti statinį pradinį puslapį, šis šablonas yra nau-	
	dojamas atvaizduoti paskutinius sukurtus įrašus.	
404.php	404 šablonas yra naudojamas, kai "WordPress" neranda įrašų, puslapio arba	
	kito turinio, kuris atitiktų lankytojo užklausą.	
search.php	Paieškos rezultatų šablonas yra naudojamas atvaizduoti lankytojo paieškos re-	
	zultatams.	

2 lentelė. "WordPress" pirminiai šabloniniai failai ir jų paskirtis. [Worc]

Dalinis šablonas yra daugkartinio panaudojimo šabloninis failas, kuris yra įtraukiamas į kitus šablonus. Dažniausiai naudojami daliniai šablonai [Worc]:

- *header.php* generuoti svetainės puslapių antraštėms;
- *footer.php* generuoti svetainės puslapių poraštėms;
- *sidebar.php* generuoti svetainės puslapių šoninei juostai.

Taip pat, galima sukurti tinkintus (angl. *custom*) dalinius šablonus ir juos įtraukti į kitus šabloninius failus [Worc].

Temos funkcijų "functions.php" failas skirtas kurti temai unikaliems metodams, kurie gali būti naudojami kaip gaudyklė (angl. *hook*) pagrindinėms "WordPress" funkcijoms. Šis failas elgiasi kaip "WordPress" įskiepis, kuris prideda svetainės funkcionalumą ir gali būti naudojamas iškviesti "WordPress" funkcijoms arba iškviesti savą (angl. *native*) PHP kodą [Worc].

Temos funkcijų faile galima [Worc]:

- naudoti "WordPress" gaudykles;
- įjungti norimą funkcionalumą;
- aprašyti funkcijas, kurias vėliau bus galima panaudoti skirtinguose šabloniniuose failuose.

3.1.2. Temos funkcionavimas

Norint išgauti sukurtus duomenis ir juos atvaizduoti "WordPress" internetiniame puslapyje yra naudojamos šablonų žymės. Jos yra skirtos nuskaityti duomenims iš duomenų bazės. Duomenys gali būti bet kas nuo įrašo pavadinimo iki pilnos įrankių juostos [Worc].

Šablono žymė yra programinis kodas (funkcija), kuris yra išskirtas į tris komponentus [Worc]:

- PHP kodo žymę;
- "WordPress" funkciją;
- neprivalomus parametrus.

Pavyzdžiui, šablono žymė "get_header()" nurodo, kad reikia nuskaityti antraštės "header.php" failą ir įtraukti jį į šią žymę kviečiantį temos šabloną.

Taip pat, "WordPress" temų kūrimo procesui yra svarbus ciklo principas, kuris yra naudojamas atvaizduoti įrašams temos šabloniniuose failuose. Kiek įrašų reikia atvaizduoti yra nusprendžiama pagal nustatymą "WordPress" konsolėje. Ciklo viduje, "WordPress" paima kiekvieną įrašą, kurį reikia atvaizduoti puslapyje, ir suformatuoja jį priklausomai nuo temos nurodymų [Worc].

Ciklas išskiria duomenis kiekvienam įrašui iš "WordPress" duomenų bazės ir įtraukia reikiamą informaciją į šabloninę žymę. Jis gali turėti skirtingus panaudojimo būdus, pavyzdžiui [Worc]:

- atvaizduoti įrašų pavadinimus ir ištraukas tinklalapio pradiniame puslapyje;
- atvaizduoti turinį ir komentarus vieno įrašo puslapyje;
- atvaizduoti turinį individualiame puslapyje, naudojant šablonines žymes.

Taigi, naudojant "WordPress" šablonus ir duomenis atvaizduoti skirtas šablonines žymes, galima sukurti šios TVS temas.

3.2. Temų generavimo įrankių kriterijai

Bendras temų generavimo įrankių bruožas yra tai, kad jie padeda naudotojams lengvai kurti ir redaguoti temas, įterpti sudėtingus elementus. Pagrindiniai įrankių kriterijai yra tokie [Ger13]:

- 1. Galimybė tinkinti statinius elementus.
- 2. Galimybė tinkinti elementų išdėstymą.
- 3. Galimybė tinkinti dinaminius elementus.

Įrankiai turėtų būti tinkami naudoti pritaikytai auditorijai. Jeigu įrankis skirtas naudotojams, kurie neturi techninių žinių, tai naudotis reikalingos techninės žinios turi būti mažos ir įrankiu turi būti paprasta naudotis. Taip pat, sukurta tema turi atitikti generavimo priemonėje nurodytas savybes [Ger13].

3.3. Egzistuojantys temų generavimo įrankiai

Norint sukurti pasirinktos TVS šabloną reikia susipažinti su jos temų kūrimo principais: failų struktūra, reikiama konfigūracija, turinio valdymo sistemos programine sąsaja ir naudojama programavimo kalba. Tam, kad nereiktų gilintis į kiekvienos TVS temų kūrimo principus, galima naudoti vizualinius šablonų generavimo įrankius, kurie suteikia priemones kurti temas neturint žinių apie pasirinktos sistemos temų kūrimo detales. Internete galima rasti įvairių įrankių TVS temoms generuoti. Dauguma iš jų yra skirti tik konkrečios vienos turinio valdymo sistemos temų generavimui ir veikia ne kaip atskira programinė įranga, bet kaip redaguojama TVS tema arba įskiepis [Ger13].

Egzistuojantys įrankiai šiam darbui pasirenkami pagal jų populiarumą. Atsižvelgus į "Wordpress" temų generavimo įrankių apžvalgas, pastebėta, kad dažniausiai minimos šios generavimo priemonės [New20a; Tea20; Wpb20]:

- "Thrive Theme Builder";
- "Divi";
- "Elementor Pro".

Pasirinkti temų generavimo įrankiai yra vertinami pagal kriterijus, aprašytus 3.2 poskyryje. Norint juos įvertinti pagal aprašytus kriterijus, reikia praktiniu būdu išmėginti kiekvieną įrankį. Taigi, su kiekvienu įrankiu yra sukuriama tema, kuri turi tokias dalis:

- antraštė;
- navigacinis meniu;
- turinys;
- šoninis meniu;
- poraštė.

Šios dalys pasirinktos todėl, kad tai yra minimalios pagrindinės internetinio puslapio dalys [Con]. Išmėginus kiekvieną įrankį, jis yra įvertinamas pagal aprašytus kriterijus.

3.3.1. "Divi"

"Divi" yra mokamas nutempimo principu veikiantis (angl. *drag and drop*) svetainių šablonų kūrimo įrankis, naudojamas kaip "Wordpress" įskiepis. Su juo galima suformuoti svetainę ir redaguoti visas jos dalis, įskaitant antraštę, poraštę ir turinio (angl. *body*) išdėstymą. "Divi" galima naudoti dviejose formose: kaip *back-end* daryklę per "Wordpress" prietaisų skydelį, kartu su visais kitais "Wordpress" nustatymais, arba, kaip vizualinę daryklę, tiesiogiai redaguojamoje svetainėje. Esminis šių formų skirtumas yra naudotojo sąsaja. *Back-end* daryklėje elementai yra išdėstyti kaip blokai. Vizualinė daryklė leidžia kurti ir redaguoti puslapį internetinio puslapio *front-end* kartu su naudotojo sukurtais duomenimis. Šiame darbe yra išmėginta *back-end* daryklė [Div]. Daryklė naudoja tris pagrindinius kūrimo blokus: skyrius, eilutes ir modulius. Naudojant šiuos blokus galima sukurti daug puslapių maketų. Skyriai yra didžiausias kūrimo blokas ir juose yra laikomos eilučių grupės. Eilutės yra naudojamos moduliams talpinti, moduliai dedami į eilučių vidų. Tai yra kiekvienos "Divi" svetainės struktūra [Div].

Daryklė siūlo redaguoti tris standartines puslapio dalis: antraštę, turinį ir poraštę. Kiekvienai iš šių dalių yra skirtas atskiras redaktorius, kuriame galima nustatyti pasirinktos dalies išdėstymą, pridėti tokius modulius, kaip mygtuką, garso įrašą, nuotrauką, žemėlapį, tekstą ar kita. Pasirinkus modulį, vaizduojamas jo redagavimo langas, kuriame galima keisti modulio dizainą, turinį ir įtraukti tinkintą dokumento vaizdavimui redaguoti skirtą "CSS" kalbą. Redaguojant elementus nereikia rašyti programinio kodo – viskas yra atliekama sąveikaujant su vartotojo sąsaja.

Atliekant pakeitimus, galima stebėti, kaip jie atrodo skirtinguose ekranų dydžiuose. Išdėstymui skirtų formų skaičius yra baigtinis, tačiau jas jungiant viena su kita, galima išgauti bet kokią norimą formą. Kiekvieną elementą galima kurti nuo pradžių, tačiau galima pasirinkti jau egzistuojantį, ir jį pritaikyti savo temoje [Div].

"Divi" yra patogus naudoti temų generavimo įrankis, kuris leidžia sukurti bet kokio tipo internetinį puslapį. Tačiau verta pastebėti, kad dėl didelio elementų kiekio, redagavimo galimybių ir siūlomo funkcionalumo, gali prireikti laiko suprasti kaip šiuo įrankiu naudotis, bei išmokti visas jo galimybes.

3.3.2. "Elementor Pro"

"Elementor Pro" yra mokama nutempimo principo puslapių daryklė, kuri veikia kaip "Wordpress" įskiepis. Ji leidžia susikurti šablonus, kurie gali būti pritaikyti tiek atskiroms turinio dalims, tiek bendrai, pavyzdžiui, visiems įrašams. Kaip ir su kitomis daugiausiai naudojamomis temų kūrimo priemonėmis, "Elementor Pro" temoms kurti nereikia programavimo žinių, nes elementai ir išdėstymas gali būti sudėtas naudojantis vartotojo sąsaja [Ele; K20].

Puslapiui kurti yra naudojama tokia struktūra [K20]:

- puslapis gali turėti bet kokį kiekį sekcijų;
- kiekviena sekcija gali turėti keletą stulpelių;
- kiekvienas stulpelis gali turėti keletą turinio elementų;
- kiekvienas turinio elementas gali būti individualiai stilizuojamas.

Turinio elementų yra įvairių, galima pridėti nuotraukas, vaizdo įrašus, mygtukus, žemėlapius, formas ir kita. Suteikiamas elementų sąrašas yra gana panašus į "Divi" įrankio modulius. Kiekvienam elementui galima pakeisti jo turinį, stilių, bei pridėti pažengusias funkcijas, pavyzdžiui, CSS stilių. Taip pat, sukurtą stilių peržiūrėti skirtinguose ekrano formatuose. Skirtingai, nei "Divi" įrankyje, stilizavimo juosta yra rodoma visada toje pačioje puslapio vietoje. Taip pat, įrankyje veikia dešinio paspaudimo meniu, kuris leidžia vykdyti veiksmus su pasirinktu elementu [New20b].

Su įrankiu galima sukurti atskirus šablonus, kurie yra pritaikomi pagal nurodytas kiekvieno šablono taisykles. Pavyzdžiui, sukūrus antraštės šabloną, galima pasirinkti, ar jis turi būti matomas visuose puslapiuose. Taip pat, tokius šablonus galima kurti kitoms standartinėms "Wordpress" temos dalims: įprastam puslapiui, skyriui, nukreipimo puslapiui, iššokančiam langui, antraštei, poraštei, vienam įrašui, vienam puslapiui, archyvui, paieškos rezultatui ir klaidos puslapiui.

3.3.3. "Thrive Theme Builder"

"Thrive Theme Builder" yra taip pat mokama nutempimo principu veikianti "WordPress" redaguojama tema. Šios kūrimo priemonės išskirtinumas yra tai, kad ji automatiškai pritaiko paprastą temą turiniui atvaizduoti ir naudotojas vėliau ją gali redaguoti. Dėl šios priežasties, įrankiu sukurtuose "WordPress" internetiniuose puslapiuose dažnai vyrauja panaši estetika [New20a; The].

Pradedant naudotis šia temų kūrimo priemone, iš pradžių yra konfigūruojama tokios struktūros tema: svetainės logotipas, vyraujanti svetainės spalva, antraštė, poraštė, pradinis puslapis, vieno įrašo puslapis, įrašų sąrašas ir įprastas puslapis. Nustačius šiuos tipus, svetainė yra paruošta naudoti. Taip pat, šiuos šablonus galima redaguoti vaizdinėje vartotojo sąsajoje, kur yra vaizduojami naudotojo sukurti duomenys "WordPress" konsolėje. Tačiau yra galimybė temą redaguoti ir su įrankio sugeneruotais duomenimis, kas yra patogu, jei naudotojas internetiniame puslapyje dar nėra pridėjęs norimo turinio [The].

Įrankis leidžia redaguoti visus komponentus, o jų išsidėstymą galima stebėti skirtinguose ekrano dydžiuose. Tai galima padaryti per naudotojo sąsają arba pridedant tinkintą "CSS" stilių. Pavyzdžiui, įrašui atvaizduoti galima pasirinkti įrašo rodomą nuotrauką, autoriaus vardą, įrašo paskelbimo datą, kategorijas ir kita. Redaguojant puslapius, reikia pasirinkti kurias puslapio dalis norima rodyti. Taip pat, norint pakeisti puslapio elementų išsidėstymą, juos reikia nutempti į norimą vietą, skirtingai nei kituose įrankiuose, kur išsidėstymui yra pasirenkamos atskiros formos ir tada į jas dedami elementai.

3.3.4. Egzistuojančių temų generavimo įrankių palyginimas

Visi nagrinėti įrankiai suteikia galimybę pilnai modifikuoti "WordPress" puslapio atvaizdavimą ir jie atitinka visus apibrėžtus temų generavimo įrankio kriterijus (3.2 poskyris): su jais galima keisti statinius ir dinaminius elementus, bei jų išsidėstymą. Taip pat, jais naudojantis nereikia turėti temos kūrimui reikalingų programavimo žinių, jie prisitaiko prie skirtingų ekrano dydžių ir atvaizduoja puslapį taip, kaip yra nurodyta kuriant temą. Visi nagrinėti įrankiai suteikia jau sukurtus šablonus ar šablonų dalis, kurias naudotojas gali išmėginti arba naudoti savo internetiniame puslapyje. Tačiau esant poreikiui, visas temos dalis galima susikurti nuo pradžių.

Nagrinėjant pasirinktas priemones, pastebėti tokie temų kūrimo skirtumai (3 lentelė):

	"Divi"	"Elementor Pro"	"Thrive Theme Builder"
Įrankio tipas	Konfigūruojama tema	Įskiepis	Konfigūruojama tema
Elementų tinkinimas	Per vartotojo sąsają ir	Per vartotojo sąsają ir	Per vartotojo sąsają ir "CSS"
	"CSS"	"CSS"	
Kūrimo principas	Elementų nutempimas	Elementų nutempimas	Elementų nutempimas
Temos struktūra	Šablonai	Šablonai	Šablonai
Šablonų struktūra	Skyriai, eilutės, moduliai	Sekcijos, stulpeliai, ele-	Viršus, turinys, apačia
		mentai	
Šablonų rūšys	Pradinis puslapis, speci-	Puslapis, iššokantis	Antraštė, poraštė, pradinis
	finis puslapis, visi įra-	langas, antraštė, poraštė,	puslapis, vieno įrašo pusla-
	šai, specifinis įrašas, pa-	vienas įrašas, archyvas,	pis, įrašų sąrašas, įprastas
	ieškos rezultatai, klaidos	paieškos rezultatas,	puslapis ir kiti
	puslapis ir kiti	klaidos puslapis ir kiti	
Elementų pavyzdžiai	Nuotrauka, navigacinis	Tekstas, nuotrauka, myg-	Tekstas, nuotrauka, stulpe-
	meniu, įrašo atributai,	tukas, įrašai, navigacinis	liai, mygtukas, lentelė, že-
	tekstas, paieškos laukas,	meniu, formos, galerija ir	mėlapis ir kita
	komentarai ir kita	kita	

3 lentelė. "Divi", "Thrive Theme Builder", "Elementor Pro" savybės [Div; Ele; New20b; The]

3.4. "WordPress" temų generavimui pritaikytas "Blockly" įrankis

Nustačius "WordPress" temų generavimo įrankio kriterijus, bei išnagrinėjus egzistuojančias kūrimo priemones ir jų savybes, "Blockly" įrankį temų generavimui galima išbandyti praktiniu būdu. Šiame poskyryje yra aprašomas sukurtas įrankis.

3.4.1. Temų generavimui pritaikyta vaizdinė kalba

"Wordpress" temos pagrindas yra šabloniniai failai. 3.3 poskyryje nagrinėti įrankiai siūlo naudotojui kurti skirtingas šablonų rūšis, tačiau tam, kad su įrankiu būtų galima sukurti bent pagrindinį temos funkcionalumą, reikia generuoti šablonus, kurie yra apibrėžti 3.1.1 poskyryje. Dėl to yra sukuriami blokai, kurie atvaizduoja šiuos šablonų tipus. Kadangi kiekvienas blokas atvaizduoja skirtingą šabloninį failą, yra svarbu juos pavadinti taip, kad jie būtų atpažįstami programiniame kode, nes kiekvieno bloko kodas bus generuojamas į atskirą failą "WordPress" temoje. Taigi, yra sukuriami tokie blokai kaip 4 pav. Šių blokų paskirtis yra talpinti ir sujungti šablono elementus, kuriuos naudotojas gali pridėti kaip antrinius (angl. *child*) blokus, todėl jie negeneruoja papildomos logikos. Blokams priskyrus vaizdavimo stilių, ant elemento yra uždedamas įterptasis stilius (angl. *inline CSS*).



4 pav. Pagrindiniai šabloniniai blokai

Prie šablonų blokų yra priskirti ir 3.1.1 poskyryje aprašyti daliniai šablonai: antraštė ir poraštė. Įprastai daliniuose šablonuose yra talpinamas navigacinis meniu, kuris gali būti sukurtas "WordPress" konsolėje, todėl šių blokų kodas turi talpinti "WordPress" funkcijas. Norint tinkinti navigacinį meniu, kviečiamai funkcijai reikia nurodyti stiliaus klasę iš stilių lentelės failų. Dėl šios priežasties, "CSS" stilius poraštei ir antraštei privalo būti generuojamas į atskirą "CSS" failą. Standartiniu "Blockly" kodo generavimo būdu to padaryti negalima, nes generuojant visą temą, yra žinomas tik pagrindinių šablonų blokų kodas, kuriame jau patalpintas antrinių blokų kodas. Dėl to, nėra aišku kurį kodą reikia generuoti į atskirus failus. Įrankyje įgyvendintas sprendimas yra talpinti kodą į failą tiesiai iš bloko aprašymo funkcijos.

Šio sprendimo privalumas yra tai, kad nereikia tėviniams blokams perduoti į "CSS" failą talpinti reikiamo kodo. Šio sprendimo trūkumai yra:

- bloko kodo funkcija gauna papildomą atsakomybę generuoti kodą į failą. Pažeidžiamas vienos atsakomybės principas;
- norint panaikinti į failą sugeneruotą kodą reikia iš naujo sukurti bloką.

"WordPress" dokumentacijoje nurodyta, kad tema keičia svetainės dizainą, dažniausiai įtraukiant svetainės puslapių elementų išsidėstymą (3.1 skyrius). Taip pat, 3.2 poskyryje nurodytuose kriterijuose yra pabrėžiama, kad turi būti galimybė tinkinti elementų išdėstymą, todėl temos generavimo įrankyje turi būti galima jį keisti. Pagal 3.3.4 poskyrio 3 lentelę matoma, kad nagrinėtų temų generavimo priemonių išdėstymo keitimo būdai yra skirtingi.

Norint konfigūruoti elementų išsidėstymą, galima naudoti "CSS" taisykles. Tačiau egzistuoja jau sukurtų karkasų, kurie "CSS" ir "JavaScript" pagrindu aprašo komponentus, taip palengvindami programuotojo darbą. Vienas iš žinomiausių tokių karkasų yra "Bootstrap", kuris skirtas mobiliesiems ekranams pritaikytoms svetainėms kurti, nes automatiškai reaguoja į kintančius ekrano dydžius. Šis karkasas aprašo komponentus, kurie naudojami kaip HTML elementų klasės ir suteikia šiems elementams stilius. "Boostrap" išdėstymui naudoja eilučių ir stulpelių principą, todėl įrankyje išdėstymui redaguoti pasirinkti du blokai: eilutė (5 pav. 1) ir stulpelis (5 pav. 2). Išdėstymo blokai yra kaip įdėklas (angl. *wrapper*), skirtas nurodyti į jį įdėtų antrinių blokų išdėstymą puslapyje [Boo].



5 pav. Išdėstymo blokai eilutė (angl. row) ir stulpelis (angl. column)

Eilutės ir stulpelio blokai leidžia į save talpinti antrinius blokus. Taip pat, stulpelio blokui reikia nurodyti jo plotį (angl. *Width number*). Jis yra reguliuojamas pagal "Bootstrap" sistemą - daugiausiai leidžiamas plotis yra 12 ir visi stulpeliai sudėjus vienoje eilutėje turėtų daugiausiai sudaryti dvylikos taškų plotį [Boo].

Turint pagrindinius ir antrinius šablonus, bei išdėstymui modifikuoti skirtus blokus, galima kurti antrinius blokus, kurie atvaizduoja temos elementus. Jie yra talpinami išsidėstymo ir pirminių blokų viduje. Jų aibė gali būti didelė – vieni iš galimų elementų yra aprašyti 3.3.4 poskyrio 3 lentelėje. Įrankyje pasirinkta įgyvendinti keletą tokių elementų. Jie pavaizduoti 6 pav.





Sukurti blokai, jų paskirtis ir generuojamas kodas aprašyti 4 lentelėje:

Blokas	Paskirtis	Generuojamas kodas
Priskirto turinio	Skirtas atvaizduoti "WordPress" konsolėje tam šablonui	"WordPress" funkcijos
(6 pav. 1)	priskirtam turiniui. Kuriant pirminius blokus ir generuo-	"PHP" kalba, skirtos atvaiz-
	jant jų kodą, yra nurodomas šablono pavadinimas, kuriam	duoti "WordPress" duomenų
	galima priskirti turinį "WordPress" konsolėje. Jeigu į pir-	bazėje laikomą puslapiui
	minį bloką pridėsime priskirto turinio bloką, svetainėje	priskirtą turinį.
	bus atvaizduotas konsolėje priskirtas turinys.	
Paieškos lauko	Skirtas atvaizduoti paieškos lauko elementą. Paieškos	Generuoja "WordPress"
(6 pav. 2)	laukui yra sukurtas atskiras blokas tam, kad būtų galima	funkciją "PHP" kalba, skirtą
	pamatyti sukurtą paieškos šabloną search.php. Sukūrus	paieškos lauko šablonui
	paieškos šabloną ir pridėjus paieškos lauką, į paieškos lau-	atvaizduoti.
	ką įvedus reikšmę yra automatiškai atvaizduojami paieš-	
	kos rezultatai.	
Įrašų (6 pav. 3)	Skirtas atvaizduoti "WordPress" konsolėje sukurtiems	Generuoja "WordPress"
	įrašams. Jam galima nurodyti įrašų per puslapį skaičių,	funkcijas "PHP" kalba, skir-
	nustatyti ar rodyti miniatiūrą, taip pat sukonfigūruoti kaip	tas atvaizduoti "WordPress"
	įrašai bus atvaizduojami, pridedant turinio blokus ir pri-	duomenų bazėje laikomus
	dėti įrašo atvaizdavimo stilių.	įrašus.
Archyvo (6 pav.	Skirtas atvaizduoti archyvo arba paieškos rezultatų įra-	Generuoja "WordPress"
4)	šams. Šis blokas turėtų būti naudojamas tik su Archive ar-	funkcijas "PHP" kalba, skir-
	ba Search pirminiais blokais/šablonais. Siame bloke ga-	tas atvaizduoti "WordPress"
	lima nurodyti kiekvieno atvaizduoto įrašo atvaizdavimo	duomenų bazėje laikomus
	stilių.	įrašus, atitinkančius archyvo
		ar paieškos užklausas.
Meniu (6 pav. 5)	Skirtas atvaizduoti "WordPress" konsolėje sukurtus me-	Generuoja "WordPress"
	niu. Bloke galima pasirinkti, kurį meniu atvaizduoti ir	funkcijas "PHP" kalba,
	prideti teksto stilių.	skirtas atvaizduoti meniu
		ir "CSS" failą, kuriame
		talpinamas meniu sąraso
D	<u>Öltra and et il attentione ti iltitetti Ötra illi iltitetti ö</u>	Constant LITMU "
Paveiksiello (6	Skirtas atvaizduoti paveiksieliui. Siam biokui reikia nu-	Generuoja "HIML" zymę
pav. o)	rodyti paveiksieno universaiųjį isteknaus adresą (URL),	paveiksienui atvaizduoti.
Takata (6 nov. 7)	Jo skalurullią, auksų ir pioų.	Canamia UTMI " ¥
1eksto (o pav. 7)	skirtas pavaizduoti sutormatuotą tekstą. Teksto blokuli golima priekirti įvairius stilius su taksta stilius konfoji	teketui etveizdueti
	ganna priskiru įvairus sunus su teksto sunaus konfigu-	tekstul atvalzduoli.
	racijos bioku ir nurodyti teksto elemento parastes.	

4 lentelė. Antriniai blokai, jų paskirtis ir generuojamas kodas.

Norint konfigūruoti atskiras įrašų dalis, reikia sukurti tokios paskirties blokus. Tam sukurti turinio elementų blokai, kurie skirti atvaizduoti archyvo arba paieškos rezultatų įrašams ir įprastiems "WordPress" konsolėje sukurtiems įrašams. Tokie blokai yra: įrašo pavadinimas (7 pav. 1), įrašo turinys (7 pav. 2) ir įrašo ištrauka (7 pav. 3).



7 pav. Įrašų elementų blokai

Įrašų elementų blokai generuoja atitinkamas "WordPress" funkcijas "PHP" kalba, kurios atvaizduoja įrašo pavadinimą, turinį arba ištrauką. Jeigu yra pridėtas elemento teksto stilius, šis stilius yra prijungiamas prie atvaizduojamo elemento, kaip įterptasis stilius.

Temos elementų konfigūravimas, kaip stiliaus, spalvų, rėmelių ir kitų savybių keitimas yra neatsiejama temos kūrimo dalis. 3.2 poskyrio kriterijuose nurodyta, kad turi būti galimybė tinkinti tiek statinius, tiek dinaminius elementus. Tokiai konfigūracijai yra reikalingas "CSS" stilius arba jį apgaubiantys karkasai. Įrankyje yra įgyvendinti keletas konfigūracijos blokų, kaip pavaizduota 8 pav.



8 pav. Konfigūraciniai blokai

Sukurti bl	okai, jų	paskirti ir	generuojamas	kodas:
	1 1 6	1	0 1	

Blokas	Paskirtis	Generuojamas kodas	
Stiliaus (8 pav. 1)	Stiliaus blokas yra skirtas sujungti visus ki-	Apjungia visų prie jo prijungtų blo-	
	tus konfigūracinius blokus, tam, kad vie-	kų kodą.	
	nam stiliaus laukeliui būtų galima priskirti		
	įvairius stiliaus elementus.		
Dydžio (8 pav. 2)	Šis blokas yra skirtas nurodyti elemento il-	Generuoja "CSS" kodą, kuris skir-	
	giui ir pločiui. Prie šio bloko prijungiami	tas nurodyti ilgį ir plotį.	
	procentinio ir taškų dydžio blokai.		
Paraščių ir užpildy-	Šie blokai skirti kontroliuoti elementų išsi-	Generuoja "CSS" kodą, kuris skir-	
mo stiliaus (8 pav. 3	dėstymą šablonuose. Jie nurodo elementų	tas paraštėms arba užpildymui reda-	
ir 7)	paraštes ir užpildymą.	guoti.	
Dydžio matmenų (8	Skirti naudoti ten, kur kituose konfigūraci-	Generuoja įvestą skaičių kartu su	
pav. 6)	niuose blokuose yra prašoma nurodyti dy-	procentiniu arba taškų dydžiu.	
	dį.		
Teksto stiliaus (8	Tai yra teksto stiliaus konfigūracijos blo-	Generuoja tekstui atvaizduoti skirtą	
pav. 4)	kas. Jis leidžia nurodyti šriftų šeimą, šrifto	"CSS" kodą.	
	stilių, šrifto svorį, dydį, spalvą, fono spalvą,		
	lygiavimą ir teksto transformaciją.		
Fono spalvos (8 pav.	Prijungus spalvos pasirinkimo bloką, yra	Generuoja "CSS" kodą, kuris skir-	
5)	sugeneruojama elemento spalva.	tas nurodyti fono spalvą.	
Rėmelio (8 pav. 8)	Šis blokas yra skirtas uždėti rėmeliui ant	Generuoja rėmeliui atvaizduoti skir-	
	norimo turinio.	tą "CSS" kodą.	

5 lentelė. Konfigūraciniai blokai, jų paskirti ir generuojamas kodas.

Tokių sukurtų blokų užtenka tam, kad būtų sugeneruojama bazinio funkcionalumo "WordPress" tema. Blokų jungimo pavyzdžiai ir jų generuojamas kodas yra patalpintas 2 priede. Visos sukurtos sistemos programinis kodas yra 1 priede.

3.4.2. Naudotojo sąsaja

"Wordpress" temų generavimui pritaikytas "Blockly" įrankis yra naudojamas per internetinę naršyklę - jis yra patalpintas "HTML" faile. Šiame faile yra:

- "Blockly" darbalaukis su naudoti parengtais blokais;
- kodo sritis, kuri darbalaukyje jungiant blokus atvaizduoja koks kodas yra generuojamas;
- tiesiogiai atvaizduojamo "Wordpress" puslapio sritis, kurioje galima matyti, kaip jungiami blokai ir generuojami šablonai atrodo "Wordpress" svetainėje;
- mygtukas, skirtas sugeneruoti temos archyvą (angl. *zip*) tam, kad sudėjus šablonus, juos būtų galima įkelti kaip temą į "Wordpress" konsolę.

"Blockly" darbalaukyje yra sukurti blokų jungimo pavydžiai. Tai reiškia, kad atsidarius blokų pasirinkimo įrankių juostą, galima pasirinkti ne tik pavienius blokus, bet ir sudėtas blokų dalis. Tai gali padėti suprasti, kaip reikia jungti blokus ir pagreitinti temos kūrimo procesą. Kodo sritis yra skirta stebėti, kokį kodą generuoja sudėti blokai. Atlikus bet kokį pakeitimą "Blockly" darbalaukyje, kodo sritis yra atnaujinama. Taigi, visada yra matomas naujausias generuojamas kodas. Tai gali padėti naudotojui suprasti, koks kodas yra generuojamas iš dedamų blokų ir ar būtent toks kodas yra reikalingas.

Nagrinėjant egzistuojančius temų generavimo įrankius, pastebėta, kad jie veikia nutempimo principu "WordPress" aplinkoje, pavyzdžiui, kaip redaguojama tema arba įskiepis (3 lentelė). Vadinasi, kad atliekant pakeitimus, yra iš karto matoma, kaip kuriama tema atrodys internetiniame puslapyje. Šiame darbe kuriamas "Blockly" įrankis veikia kaip atskira internetinė svetainė ir nėra "WordPress" papildinys, tačiau norint jį integruoti kartu su "WordPress", galima tokį įrankį kurti ne kaip atskirą svetainę, o kaip "WordPress" įskiepį [Wora].

Kadangi "Blockly" blokams kurti reikalinga šio įrankio darbo sritis, nutempimo principu veikiančio įrankio sukurti nepavyks. Taigi, norint tiesiogiai atvaizduoti, kaip jungiami blokai atrodo internetiniame puslapyje, reikia ieškoti kitų būdų nei nutempimo principas. Šiame darbe tiesiogiai atvaizduojamo "Wordpress" puslapio sritis yra rodoma "HTML" elemente "iframe", kuris nurodo į lokaliai patalpintą "Wordpress" puslapį. Kai naudotojas pilnai sudeda darbalaukyje esančius blokus, asinchroniniu būdu yra siunčiama užklausa į lokalų serverį. Lokalus serveris, parašytas su "NET Core" karkasu, turi programų sąsąją su valdikliu, skirtu atnaujinti lokaliai patalpinto "Wordpress" puslapio temą. Užklausoje yra priimamas failo pavadinimas ir failo turinys. Pagal failo pavadinimą lokalaus "Wordpress" puslapio kataloge yra pridedamas arba atnaujinamas failas su pateiktu failo pavadinimu. Atsiųstas failo turinys yra įrašomas arba perrašomas ant jau egzistavusio turinio. Tuomet, jeigu operacija įvykdyta sėkmingai, yra grąžinamas sėkmės rezultatas. Gavus sėkmės rezultatą, įrankyje atnaujinama "iframe" atvaizduojama nuoroda ir yra matomi sukurtų blokų pakeitimai "Wordpress" puslapyje [Doca].

3.5. Gautų rezultatų apibendrinimas

Šiame skyriuje buvo aprašytas įrankis, kuris pritaiko "Blockly" blokais paremtą programavimo kalbą "WordPress" temų generavimui. Jis leidžia generuoti bazinio funkcionalumo temas. Įrankis įgyvendintas remiantis nustatytais temų generavimo kriterijais ir atlikta egzistuojančių temų generavimo įrankių analize. Įgyvendintas įrankis atitinka apibrėžtus temų generavimo įrankių kriterijus (3.2 poskyris). Konfigūraciniai blokai leidžia tinkinti statinius ir dinaminius elementus, išdėstymo blokai keičia elementų išdėstymą, o sujungus temai reikiamus blokus ir ją sugeneravus, ji atitinka nurodytas savybes.

"Blockly" suteikia sąsają, kurioje galima aprašyti norimus sukurti blokus ir jis juos automatiškai sugeneruoja. Tai yra patogu, nes nereikia įgyvendinti pačių blokų atvaizdavimo logikos ir visas dėmesys yra išlaikomas į blokų struktūrą ir jų generuojamą kodą.

Kuriant šiai sričiai pritaikytą vaizdinę programavimo kalbą pastebėta, kad "Blockly" įrankio naudojimas riboja vaizdinės kalbos integravimo aplinką. Taip yra todėl, kad įrankis turi savo darbo sritį, kuri reikalinga norint naudotis sukurtais blokais. Tai reiškia, kad nutempimo principu veikiančio įrankio, kuris yra įgyvendintas nagrinėtuose temų generavimo įrankiuose, sukurti nepavyks. Taip pat, pastebėta, kad naudojantis "Blockly" suteiktomis galimybės, yra sudėtinga tinkinti "WordPress" elementus, kuriems reikia naudoti stilių lenteles. Taip yra todėl, kad generuojant sudėtų blokų kodą, jie "Blockly" įrankio pagalba automatiškai sujungiami į programinio kodo vienetą. Dėl to, nėra aišku, kurį kodą reikia išskirti į atskirus stiliaus failus. Tačiau šiam ribojimui galima rasti sprendimų. Vienas iš jų yra elementų stiliaus blokų jungimas atskirai nuo pačių elementų.

4. "Blockly" pritaikymas TVS duomenų atvaizdavimui kurti

Antrasis su "Blockly" vykdomas bandymas yra šio įrankio pritaikymas turinio valdymo sistemų suteiktos sąsajos duomenims atvaizduoti.

Šiuo bandymu siekiama apjungti skirtingų turinio valdymo sistemų programines sąsajas ir panaudoti "Blockly" kaip priemonę iš sąsajos gautiems duomenims atvaizduoti. Sukurtos vaizdinės programavimo kalbos paskirtis yra generuoti *headless* TVS internetinius puslapius (2 skyrius). Skirtingai nuo pirmo bandymo, nėra konkrečios struktūros kaip turi būti generuojamas vaizdinės programavimo kalbos kodas. Taip pat, yra integruojama komunikacija su programine sąsaja vietoj to, kad blokai talpintų konkrečios TVS temos turiniui atvaizduoti reikalingą logiką.

Įrankiui įgyvendinti yra pasirinktos trys labiausiai naudojamos atvirojo kodo turinio valdymo sistemos: "WordPress", "Joomla" ir "Drupal" [Sur20]. Šias sistemas galima naudoti ir kaip tradicines TVS, ir kaip *headless* TVS, nes duomenis atvaizduoti galima taikyti tiek temas arba šablonus, tiek naudotis API sąsaja. Pasirinktas duomenų tipas – sistemose sukurti įrašai. Tikslas yra apjungti šiomis sistemomis sukurtas internetines svetaines, naudojantis jų programų sąsajomis išgauti duomenis apie sukurtus įrašus ir su "Blockly" sugeneruoti internetinę svetainę, kurioje bus atvaizduojami visi nauji įrašai.

Šiame skyriuje yra aprašomos naudotos programinės sąsajos, sistemos struktūra ir jos įgyvendinimo principai.

4.1. Programinė sąsaja

Programinė sąsaja (API) yra apibrėžimų ir protokolų rinkinys, skirtas kurti ir integruoti programinę įrangą. API leidžia produktui arba servisui bendrauti su kitais produktais arba servisais nežinant jų implementacijos detalių.

"WordPress" programinė sąsaja įgyvendina REST architektūrinio stiliaus savybes ir suteikia galimybę bendrauti su REST saityno paslaugomis (angl. *web services*). Kai klientas įvykdo REST API užklausą, resursų būsena yra perduodama pabaigos taškui (angl. *endpoint*). Ši informacija "WordPress" programinėje sąsajoje suteikiama JSON duomenų formatu. Kreipiantis į nurodytą pabaigos tašką yra gaunami duomenų bazėje talpinami įrašai [Hat; Worb].

"Drupal" programinė sąsaja taip pat įgyvendina REST principus. Skirtingai nei "WordPress", norint gauti duomenis iš "Drupal", visų pirma reikia duoti jų prieigą administratoriaus konsolėje ir nurodyti konkrečius duomenų tipus, kuriuos išorinis naudotojas gali pasiekti. Duomenys gali būti grąžinami JSON ir XML formatais [Dru].

"Joomla" suteikiama programinė sąsaja įgyvendina JSON: API specifikaciją. "Joomla" programinė sąsaja yra sukurta neseniai, darbo rašymo metu ji yra beta versijoje. Kaip ir "WordPress", yra suteikti duoti pabaigos taškai, kurie leidžia pasiekti skirtingų tipų duomenis. Duomenys yra grąžinami JSON formatu [API; Docb].

4.2. Sistemos struktūra

Kiekviena naudojama turinio valdymo sistema grąžina skirtingos struktūros duomenis. Tam, kad išorinės pusės dalyje nereikėtų bendrauti su skirtingais pabaigos taškais ir prisitaikyti prie jų formatų, sukurta sistemos vidinė pusė apjungia nurodytus pabaigos taškus ir konvertuoja jų duomenų tipus į vieną bendrą formatą, kurį gali naudoti išorinė pusė per suteiktą programinę sąsają. Išorinė pusė šią programinę sąsają naudoja sukurtos vaizdinės programavimo kalbos blokų kūrimui. Tuomet naudojantis "Blockly" yra jungiami blokai, kurie sugeneruoja naujus įrašus atvaizduojančią internetinę svetainę. Tokios struktūros sistema nėra apribota tik trimis turinio valdymo sistemomis. Toks įrankis gali integruoti daugiau programinių sąsajų ir perduoti jų duomenis internetinei svetainei. Sistemos struktūra pavaizduota 9 pav.



9 pav. Sistemos struktūra

Duomenų adapterio komponentas (9 pav. Duomenų adapteris) yra REST API, kuris grąžina bendros struktūros skirtingų TVS įrašus.

Turinio valdymo sistemų programinės sąsajos grąžina skirtingo formato duomenis ir juos aprašo skirtingas savybes turinčios klasės. Kadangi duomenų struktūros nėra kompleksinės ir reikiamos klasės konvertavimui užtenka paprastų konvertavimo funkcijų, sistemos duomenų adaptavimas yra įgyvendintas paveldėjimo principu, kaip pavaizduota 10 pav.



10 pav. Įrašų (angl. post) struktūros

4.3. "WordPress", "Joomla" ir "Drupal" duomenų atvaizdavimui kurti pritaikytas "Blockly"

Kuriant "WordPress" temų generavimui pritaikytą "Blockly" vaizdinę kalbą buvo pastebėta, kad blokai nėra tinkami kodui generuoti į daugiau nei vieną vietą. Todėl norint pritaikyti "CSS" stilių, jį teko rašyti kaip elemento uždedamąjį įterptą stilių. Tai nėra gera praktika, nes jį įterpiant su elemento aprašymu yra prarandami stiliaus puslapių (angl. *style sheet*) privalumai. Norint sumažinti tokio "CSS" rašymo būdo kiekį, galima taikyti karkasus, kurie suteikia paruoštus naudoti stilius. Vienas iš labiausiai naudojamų tokio tipo karkasų yra "Bootstrap". Šiame įrankyje jis yra naudojamas elementams stilizuoti [Boo; W3S].

Kadangi generuojamas internetinis puslapis neturi konkrečios nurodytos struktūros, pasirinkta kurti bazines dalis turintį puslapį. Bazines dalis sudaro antraštė, navigacinis meniu, turinys, šoninis meniu ir poraštė. Generuojamas puslapis neturi navigacijos į skirtingas svetainės vietas, todėl jam yra naudojamos tik antraštės, turinio (angl. *content*) ir poraštės dalys [Con].



11 pav. Pagrindiniai blokai

11 pav. pavaizduoti blokai generuoja bazines antraštės, turinio ir poraštės dalis. Visi kiti sukurti blokai yra skirti atvaizduoti arba konfigūruoti turinio elementus. Jie yra sudedamoji pagrindinių blokų dalis. Pozicijos nustatymui antraštės ir poraštės blokai naudoja "Bootstrap" pozicijos klases. Kiekvienam blokui galima pridėti konfigūracijos ir turinio blokus.

Išdėstymui yra sukurti blokai, kurie atitinka "Bootstrap" išdėstymo elementus: konteineris (angl. *container*), eilutė (angl. *row*) ir stulpelis (angl. *column*). Konteineris yra bazinis išdėstymo elementas ir yra privalomas naudojant įprastinę "Bootstrap" tinklelio sistemą. Konteineriai gali būti dviejų tipų: nustatyto dydžio konteineris, kurio ilgis keičiasi ties nurodytu lūžio tašku, arba,

pilno ilgio konteineris, kuris visada užima visą ekrano ilgį. Naudojant nustatyto dydžio konteinerį ir nurodant lūžio taškus, internetinis puslapis yra prisitaikantis prie skirtingo dydžio ekranų. Svarbi tinklelio išdėstymo sistemos dalis yra eilutės su stulpeliais. Jie skirti išdėstyti elementams konteineryje. Sukurti blokai pavaizduoti 12 pav. [Boo].



12 pav. Išdėstymo blokai

Išdėstymo blokų konfigūracija yra generuojama naudojantis "Bootstrap" klasėmis, kurios skirtos turinio paskirstymui eilutėje ir stulpelio pločio nustatymui pagal karkaso sistemą, kuri yra aprašyta 3.4.1 poskyryje.

Vaizdinė programavimo kalba yra taikoma konkrečiai nurodytai sričiai, todėl turinio elementai ir jiems skirti atvaizduoti blokai generuoja pagrindinę srities dalį – įrašus. Jiems atvaizduoti yra taikomas stilius ir pasirenkamas įrašų per eilutę skaičius (angl. *Posts per row*). Įrašą gali sudaryti skirtingos dalys, kaip pavadinimas (angl. *Item title*), autorius, data, turinys (angl. *Item content*), ištrauka ir kita. Šioms dalims sukurti yra skirti atskiri blokai, kurie leidžia jas konfigūruoti atskirai. Taip pat, turinio elementams atvaizduoti yra sukurti nuotraukos (angl. *image*), teksto (angl. *text*) ir mygtuko (angl. *button*) blokai. Jie gali būti pridėti bet kurioje puslapio dalyje. Sukurti blokai pavaizduoti 13 pav.



13 pav. Turinio blokai

Įrašų blokas savyje talpinta "jQuery" bibliotekos "Ajax" funkciją, kuri iškviečia nurodytą komponento pabaigos tašką ir gauna iš ten grąžinamus įrašų duomenis. Gauti duomenys yra konfigūruojami pagal sugeneruotą prijungtų stiliaus blokų kodą [jQu].

Temos elementams konfigūruoti yra taikomi tokie patys blokai, kurie yra sukurti "Wordpress" temų generavimo įrankyje. Jie aprašyti 5 lentelėje.

Sukurtų blokų užtenka sugeneruoti bazinį funkcionalumą turintį internetinį puslapį, kuris atvaizduoja skirtingų turinio valdymo sistemų įrašus. Kuriant internetinę svetainę, reikia laikytis nurodytos failų struktūros. Šiame darbe yra naudojamas HTML failas sugeneruotam programiniam kodui talpinti. Sujungus norimus blokus internetinei svetainei atvaizduoti, yra generuojamas standartinės struktūros HTML puslapis, kuriame yra pridėtos naudojamų karkasų ar bibliotekų nuorodos ir sugeneruotas kodas. Generuojamas blokų kodas nurodytas 3 priede. Jungiamų blokų pavyzdžiai ir jų generuojamas kodas pavaizduotas 4 priede [Con].

4.4. Gautų rezultatų apibendrinimas

Šiame skyriuje aprašytas sukurtas įrankis, kuris pritaiko "Blockly" vaizdinę programavimo kalbą skirtingų turinio valdymo sistemų duomenų atvaizdavimo generavimui. Duomenys atvaizduojami naudojantis suteiktomis "Joomla", "Drupal" ir "WordPress" API sąsajomis. Pastebėta, kad skirtingų TVS duomenys gana lengvai integruojasi tarpusavyje, nes jie turi panašias struktūras. Pavyzdžiui, visose sistemose kuriami įrašai su pavadinimu, turiniu, datomis ir kitais duomenimis.

Kuriant įrankį, pastebėta, kad naudojant duomenis iš programinės sąsajos, jie yra lengviau konfigūruojami, nes vaizdinė programavimo kalba gali naudotis realiais duomenimis, o ne konkrečios TVS suteiktomis funkcijomis ir tai suteikia daugiau kontrolės.

Kadangi šis įrankis buvo kuriamas po "WordPress" temų generavimui skirto įrankio, tam tikri blokai buvo panaudoti iš naujo arba pertvarkyti. Pastebėta, kad norint turėti pakartotinio panaudojamumo blokus skirtingiems komponentams, reikia, kad jie generuotų kuo mažesnę kodo dalį ir turėtų tik vieną atsakomybę. Jeigu generuojant bloko kodą reikia naudoti sąlyginį sakinį, reiškia, kad jis gali būti suskaidytas į mažesnius blokus.

Taip pat, "Bootstrap" panaudojimas blokų elementų stiliui konfigūruoti yra patogi priemonė. Taip yra todėl, kad tam tikrai atvejais galima išvengti atskiro stiliaus lentelių generavimo ir šis karkasas automatiškai palaiko skirtingų ekranų dydžius. Blokai tokiu atveju talpina mažiau logikos ir yra lengviau panaudojami.

Rezultatai ir išvados

Šio darbo tikslas buvo išnagrinėti "Blockly" įrankio, skirto kurti pritaikytas blokais paremtas vaizdines programavimo kalbas, galimybes turinio valdymo sistemų duomenų atvaizdavimui generuoti. Šių sistemų duomenų atvaizdavimas gali būti vykdomas kuriant pasirinktos TVS temas arba šablonus ir kuriant TVS suteiktos programinės sąsajos duomenų atvaizdavimo lygmenį.

Vykdant įrankio pritaikymo galimybių analizę buvo sukurtos dvi sistemos: "Blockly" vaizdinės programavimo kalbos pritaikymas pasirinktos turinio valdymo sistemos temų generavimui ir "Blockly" vaizdinės programavimo kalbos pritaikymas pasirinktų turinio valdymo sistemų suteiktos sąsajos duomenų atvaizdavimui internetinėje svetainėje. Sistemų įgyvendinimui atlikta "Blockly" įrankio elementų kūrimo analizė, nustatyti turinio valdymo sistemų bruožai ir jų panaudojimo būdai, nagrinėti "WordPress" turinio valdymo sistemos baziniai temų kūrimo principai. Siekiant sukurti pasirinktos TVS temų generavimo įrankį, buvo nustatyti tokio įrankio kriterijai ir išnagrinėta, kokiais temų kūrimo principais naudojasi jau egzistuojančios labiausiai naudojamos "WordPress" temų generavimo priemonės. Sukurtas temų generavimo įrankis atitinka nustatytus kriterijus ir naudojasi nagrinėtų įrankių generavimo principais. Taip pat, siekiant pritaikyti "Blockly" iranki įvairių TVS programinės sasajos duomenų atvaizdavimui, buvo modeliuojama sistema, kuri apjungia "Joomla", "Drupal" ir "WordPress" turinio valdymo sistemų programinių sąsajų duomenų struktūras, o tuos duomenis naudoja "Blockly" sugeneruota internetinė svetainė. Svetainei generuoti įrankyje panaudotas "Bootstrap" karkasas, kaip jungiamoji priemonė TVS duomenims atvaizduoti. Darbo metu nagrinėtos "Blockly", "WordPress", "Joomla", "Drupal" ir "Boostrap" dokumentacijos.

Vaizdinės programavimo kalbos yra taikomos įvairiose srityse ir viena iš jų gali būti internetinių svetainių generavimas arba turinio valdymo sistemų duomenų atvaizdavimas. Atlikta analizė gali padėti norint pritaikyti "Blockly" nagrinėjamoje srityje. Sukurti įrankiai yra priemonės kurti svetaines naudotojams, kurie neturi techninių žinių apie konkrečių TVS duomenų atvaizdavimo kūrimą. Taip pat, šioje srityje pritaikytą "Blockly" galima būtų panaudoti kaip mokomąją priemonę kurti turinio valdymo sistemų svetaines, kadangi blokais paremtas programavimas padeda greičiau įsisavinti ir išmokti tekstines programavimo kalbas.

Manoma, kad "Blockly" yra tinkama priemonė generuoti tradicinių ir *headless* TVS duomenų atvaizdavimo lygmenį. Tačiau darbo metu sukurti įrankiai apibrėžia tik bazinį funkcionalumą ir nėra pilnai konfigūruojami. Tęsiant šį darbą, būtų galima pilnai atsisakyti elementams generuojamo įterptojo stiliaus, pritaikyti visus analizuotuose generavimo įrankiuose naudojamus temos elementus. Taip pat, yra siūloma atsisakyti tradicinių turinio valdymo sistemų suteikiamų funkcijų duomenims išgauti ir naudoti duomenis iš TVS programinės sąsajos.

Šaltiniai

[API] JSON API. Json api specification for building apis in json. https://jsonapi.org/. Tikrinta 2020-05-15. [BGK⁺17] David Bau, Jeff Gray, Caitlin Kelleher, Josh Sheldon ir Franklyn Turbak. Learnable programming: blocks and beyond. Commun. ACM, 60(6):72-80, 2017-05. ISSN: 0001-0782. DOI: 10.1145/3015455. URL: https://doi.org/10.1145/3015455. [Blo20] Blockly. A javascript library for building visual programming editors. https:// developers.google.com/blockly, 2020. Tikrinta 2020-12-01. [Boo] Bootstrap. Bootstrap. https://getbootstrap.com/docs/4.0/layout/ overview/. Tikrinta 2020-12-01. [Con] MDN Contributors. Document and website structure. https://developer. mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_ and website structure. Tikrinta 2021-04-01. [Div] Divi. Divi builder plugin documentation. https://www.elegantthemes.com/ documentation/divi/visual-builder/. Tikrinta 2020-12-13. [Doca] MDN Web Docs. <iframe>: the inline frame element. https://developer. mozilla.org/en-US/docs/Web/HTML/Element/iframe. Tikrinta 2020-05-15. [Docb] Joomla! Documentation. Joomla api specification. https://docs.joomla.org/ Joomla_Api_Specification. Tikrinta 2020-05-15. [Dru] Drupal. Restful web services api. https://www.drupal.org/docs/drupalapis/restful-web-services-api/. Tikrinta 2020-05-15. [Ele] Elementor. Elementor features. https://elementor.com/. Tikrinta 2020-12-13. [FTV02] Filomena Ferrucci, Genoveffa Tortora ir Giuliana Vitiello. Visual programming. Encyclopedia of Software Engineering. American Cancer Society, 2002. ISBN: 9780471028956. DOI: https://doi.org/10.1002/0471028959.sof143. [Ger13] Linda Di Geronimo. Theme generator for WordPress. Magistrinis darbas, ETH Zürich, 2013. DOI: 10.3929/ethz-a-009979939. [Hat] Red Hat. What is a rest api? https://www.redhat.com/en/topics/api/whatis-a-rest-api. Tikrinta 2020-05-15. [jQu] jQuery. What is jquery? https://jquery.com/. Tikrinta 2020-05-20. [K20] Karol K. Elementor review. https://winningwp.com/elementor-review/, 2020. Tikrinta 2020-12-13. Knut Melver. Headless cms explained. https://www.sanity.io/blog/ [Mel] headless-cms-explained. Tikrinta 2020-05-15.

- [MJS14] A. Mirdha, A. Jain ir K. Shah. Comparative analysis of open source content management systems. 2014 IEEE International Conference on Computational Intelligence and Computing Research, p. 1–4, 2014. DOI: 10.1109/ICCIC.2014.7238337.
- [New20a] Colin Newcomer. 6 best wordpress theme builders 2020s. https://athemes.com/ collections/best-wordpress-theme-builders/, 2020. Tikrinta 2020-12-13.
- [New20b] Colin Newcomer. Divi vs elementor: which is the right wordpress page builder for you? https://athemes.com/reviews/divi-vs-elementor/, 2020. Tikrinta 2021-04-01.
- [PFM17] Erik Pasternak, Rachel Fenichel ir Andrew N. Marshall. Tips for creating a block language with blockly. 2017 IEEE Blocks and Beyond Workshop (B B), p. 21–24, 2017. DOI: 10.1109/BLOCKS.2017.8120404.
- [Sur20] Web Technology Surveys. Usage statistics of content management systems. https://w3techs.com/technologies/overview/content_management, 2020. Tikrinta 2020-12-01.
- [Tea20] Editorial Team. The best and most popular wordpress themes of 2021. https://www.isitwp.com/top-wordpress-themes/, 2020. Tikrinta 2020-12-01.
- [The] Thrive Themes. Introducing thrive theme builder. https://thrivethemes.com/ themebuilder/?idev_id=356. Tikrinta 2020-05-02.
- [UDi] UDig. Traditional cms vs headless cms. https://www.udig.com/diggingin/traditional-cms-vs-headless-cms/. Tikrinta 2020-05-15.
- [W3S] W3Schools. How to add css. https://www.w3schools.com/css/css_howto. asp. Tikrinta 2020-05-15.
- [Wora] WordPress. Plugin handbook. https://developer.wordpress.org/plugins/. Tikrinta 2020-05-15.
- [Worb] WordPress. Rest api handbook. https://developer.wordpress.org/restapi/. Tikrinta 2020-05-15.
- [Worc] WordPress. Theme basics. https://developer.wordpress.org/themes/ basics/. Tikrinta 2021-04-20.
- [Wor20] WordPress. Theme development. https://codex.wordpress.org/Theme_ Development, 2020. Tikrinta 2020-12-01.
- [Wpb20] Wpbeginner. 28 best wordpress multipurpose themes. https://www.wpbeginner. com/showcase/best-wordpress-multi-purpose-themes/, 2020. Tikrinta 2020-12-13.

Priedai

1 priedas. Nuoroda į "Blockly" įrankį "Wordpress" turinio valdymo sistemos temų generavimui.

https://github.com/neringga/blockly_theme_builder

2 priedas. "Wordpress" turinio valdymo sistemos temų generavimui pritaikyto įrankio sujungti blokai ir jų generuojamas kodas.



14 pav. Index šabloninio bloko pavyzdys

14 pav. generuojamas kodas:

```
color:#000000;
                        text-align:center;
                        font-style:normal;">
                Tekstas
            </div>
        </div>
        <div class="col-9">
            <?php if (have_posts()) : while(have_posts()) : the_post() ;?>
                <div style="font-size: 30px; text-align: center; margin-</pre>
bottom:20px;">
                    <?php the_title(); ?>
                </div>
            <?php the_content();?>
            <?php endwhile; endif;?></div></div>
        </div>
    <?php get_footer();?>
```



15 pav. Archive šabloninio bloko pavyzdys

15 pav. generuojamas kodas:

```
<?php /* Archive */ ?>
<?php get_header();?>
<div style="margin-right:10%; margin-left:10%; margin-top:5%; margin-
bottom:5%; background: #ff9966; border-style: solid; border-
width: 1px; border-color: #000000; ">
```

<?php if (have_posts()) : while(have_posts()) : the_post() ;?>

```
<div style=" padding-right:10px; padding-left:10px; padding-</pre>
top:10px; padding-bottom:10px;
                             ">
              <a href="<?php the_permalink();?>">
                  weight:normal; color:#000000; background-color:; text-align:left; font-
style:normal;">
                  <?php the_title();?>
                  </a>
              <div style="font-family:Arial; font-size:12px; font-</pre>
weight:normal; color:#000000; background-color:; text-align:left; font-
style:normal;">
                  <?php the_excerpt();?>
              </div>
           </div>
       <?php endwhile; endif;?>
 </div>
 <?php get_footer();?>
```

3 priedas. Nuoroda į "Blockly" įrankį TVS duomenų atvaizdavimui kurti.

https://github.com/neringga/cms-website-generator

4 priedas. TVS duomenų atvaizdavimui pritaikyto įrankio sujungti blokai ir jų generuojamas kodas.

HEADER			
Position stick	y 🔹		
Style	C	🖸 Styl	les C Background color C Hex colour code (#696969)
			C Size Width ■ Height C 50 px
Components	Cont	ainer	
	R	ow	
	A	lign left	
		Column	n
		Align 🖪	right 🔹
		Width n	number 1
		Imag	ge URL 🚺 🎸 URL 🤊
		Opa	acity 1
		Widt	th 0 40 px
		Heig	ght 0 40 px

16 pav. Antraštės bloko pavyzdys

16 pav. generuojamas kodas:

```
<header class="sticky-top" style="background: #696969;
width:;
height:50px;">
<div class="container-flex h-100">
<div class="row h-100 justify-content-start">
<div class="col-1 my-auto" style="text-align: right;">
<div class="col-1 my-auto" style="text-align: right;">
<img src="URL" alt="Image" style="width:40px;
height:40px;
opacity:1;">
</div>
</div>
```

</header>



17 pav. Antraštės bloko pavyzdys

17 pav. generuojamas kodas:

```
<div class="row" id="posts">
   <script>
       $.ajax({
        type: "get",
        contentType: "application/json; charset=utf-8",
        url: `https://localhost:44385/Posts`,
        success: function (status) {
          var code = "";
          var counter = 1;
          for (const item in status) {
            var chars = status[item].post_body.length;
            var newCode = `
            <div class="col" style="">
               <a href="${status[item].post_link}">
               ${status[item].post_title}
               </a>
               <div style="">
                  ${status[item].post_body.slice(0, 256)}
                  ${chars > 256 ? "..." : ""}
               </div>
             </div>`;
             if (counter === 1) {
```