**RĪGAS VALSTS TEHNIKUMS**

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

**KVALIFIKĀCIJAS DARBS**

**“Interneta portāls lietotāja raksta lasīšanai un ziņu izveidošanai”**

Paskaidrojošais raksts 28 lpp.

Audzēknis: Platon Martynov

Prakses vadītājs: Igors Litvjakovs

Nodaļas vadītājs: Normunds Barbāns

**Rīga 2023**

**ANOTĀCIJA**

Šajā kvalifikācijas dokumentā ir aprakstīts darbibas princips un izveidošans process priekš izveidotai vietnei ziņu un lietotāju rakstu izveidei. Sistēma sniedz iespēju veidot savus rakstus, lai dalītos ar interesantu un būtisku informāciju ar citiem vietnes dalībniekiem un lietotājiem, kā arī iepazītos ar citiem rakstiem par interesējošām tēmām. Sistēma ir izstrādāta, izmantojot ‘WordPress’ ietvaru PHP programēšanas valodā un izmanto Phpmyadmin, lai regulētu un moderēt datus datu bāzē.

Kvalifikācijas darbs satur ievadu, uzdevuma izklāstu, prasību specifikāciju, izvēlēto izstrādes tehnoloģiju aprakstu, marķēšanas termināļu sistēmas modelēšanu un projektēšanu, datu struktūras aprakstu, lietotāja rokasgrāmatu un lietojumprogrammas. Ievadā ir aprakstīti aktuāli uzdevumi un problēmas, kas saistīta ar informācijas izplatīšanai visā pasaulē, un tās piedāvātās iespējas risināt to.

Kvalifikācijas darbs sastāv no 55 lappusēm, kurā ietilpst 23 attēli, 17 tabulas un 10 pielikumus. Pielikumi satur programmas pirmkodu.

**ANOTATION**

This qualification document describes the working principle and creation process for an established website for creating news and user articles. The system provides an opportunity to create your own articles to share interesting and relevant information with other members and users of the site, as well as to familiarize yourself with other articles on topics of interest. The system is developed using the 'WordPress' framework in the PHP programming language and uses Phpmyadmin to regulate and moderate the data in the database.

The qualification work contains an introduction, statement of the task, specification of requirements, description of the selected development technologies, modeling and design of the marking terminal system, description of the data structure, user manual and applications. The introduction describes current tasks and problems related to information dissemination around the world, and the opportunities it offers to solve them.

The qualification work consists of 55 pages, which includes 23 figures, 17 tables and 10 appendices. The attachments contain the source code of the program.

SATURS

[**IEVADS 2**](#_Toc167961210)

[**1. UZDEVUMA NOSTĀDNE 3**](#_Toc167961211)

**2.** [**PRASĪBU SPECIFIKĀCIJA 5**](#_Toc167961212)

[2.1. Ieejas un izejas informācijas apraksts 5](#_Toc167961213)

[*2.1.1. Ieejas informācijas apraksts 5*](#_Toc167961214)

[*2.1.2. Izejas informācijas apraksts 5*](#_Toc167961215)

[2.2. Funkcionālās prasības 6](#_Toc167961216)

[2.3. Nefunkcionalās prasības 7](#_Toc167961217)

[**3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS 10**](#_Toc167961218)

[**4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA 12**](#_Toc167961219)

[4.1. Sistēmas struktūras modelis 12](#_Toc167961220)

[4.1.1. Sistēmas arhitektūra 12](#_Toc167961221)

[4.1.2. Entītiju relāciju datu modelis (ER-diagramm) 14](#_Toc167961222)

[4.2. Funkcionālais sistēmas modelis 16](#_Toc167961223)

[4.2.1. Datu plūsmu modelis 16](#_Toc167961224)

[**5. DATU STRUKTŪRAS APRAKSTS 20**](#_Toc167961225)

[5.1. Datu bāzes fiziskā struktūra 20](#_Toc167961226)

[**6. LIETOTĀJA CEĻVEDIS 26**](#_Toc167961227)

[6.1. Sistēmas prasības aparatūrai un programmatūrai 26](#_Toc167961228)

[6.2. Sistēmas instalācija un palaišana 26](#_Toc167961229)

[6.3. Programmas apraksts 28](#_Toc167961230)

[*6.3.1. Rakstu/ziņu veidošana 29*](#_Toc167961231)

[*6.3.2. Rakstu/ziņu lasīšana 29*](#_Toc167961232)

[*6.3.3. Rakstu/ziņu kārtošana pēc kategorijas 30*](#_Toc167961233)

[*6.3.4. Rakstu/ziņu meklēšana meklēšanas joslā 31*](#_Toc167961234)

[6.5. Testa piemērs 32](#_Toc167961235)

[**NOBEIGUMS 34**](#_Toc167961236)

[**IZMANTOTIE AVOTI 35**](#_Toc167961237)

[**PIELIKUMI 36**](#_Toc167961238)

# IEVADS

Mūsdienu pasaulē būt informētam ir svarīgāk nekā jebkad agrāk, un ziņu portāliem ir būtiska nozīme, sniedzot sabiedrībai jaunāko informāciju par vietējiem, valsts un globāliem notikumiem. Līdz ar digitālo tehnoloģiju attīstību ziņu portāli ir kļuvuši par būtisku mūsu ikdienas sastāvdaļu, piedāvājot ērtu veidu, kā piekļūt ziņām un informācijai no dažādiem avotiem.

Viena no ziņu portālu priekšrocībām ir to spēja piegādāt ziņas reāllaikā. Atšķirībā no tradicionālajiem drukātajiem medijiem, kurus ierobežo drukas grafiki, ziņu portāli var atjaunināt savu saturu notikumu attīstības gaitā, nodrošinot lietotājiem jaunāko informāciju, tiklīdz tā kļūst pieejama. Tas padara ziņu portālus par nenovērtējamu resursu ikvienam, kas vēlas būt informēts par jaunākajām ziņām un aktualitātēm.

Vēl viena ziņu portālu stiprā puse ir spēja sniegt plašu skatījumu uz ziņām. Atšķirībā no tradicionālajiem medijiem, kuru atspoguļojumu var ierobežot ģeogrāfiskās vai politiskās robežas, ziņu portāli var apkopot ziņas un informāciju no avotiem visā pasaulē. Tādējādi lietotāji var piekļūt dažādiem viedokļiem un iegūt visaptverošāku izpratni par problēmām un notikumiem, kas veido mūsu pasauli.

Pēc mūsu veiktajiem aprēķiniem, konkurentu, to stipro un vājo pušu analīzi, apkopot labāko un novērst sliktāko, mes nonācām pie secinājuma, ka ir nepieciešams izveidot ziņu portālu, kas tiks papildināts ar oriģinālām idejām un gatavs konkurēt šajā jomā ne tikai lokāli, bet arī globāli nākotne.

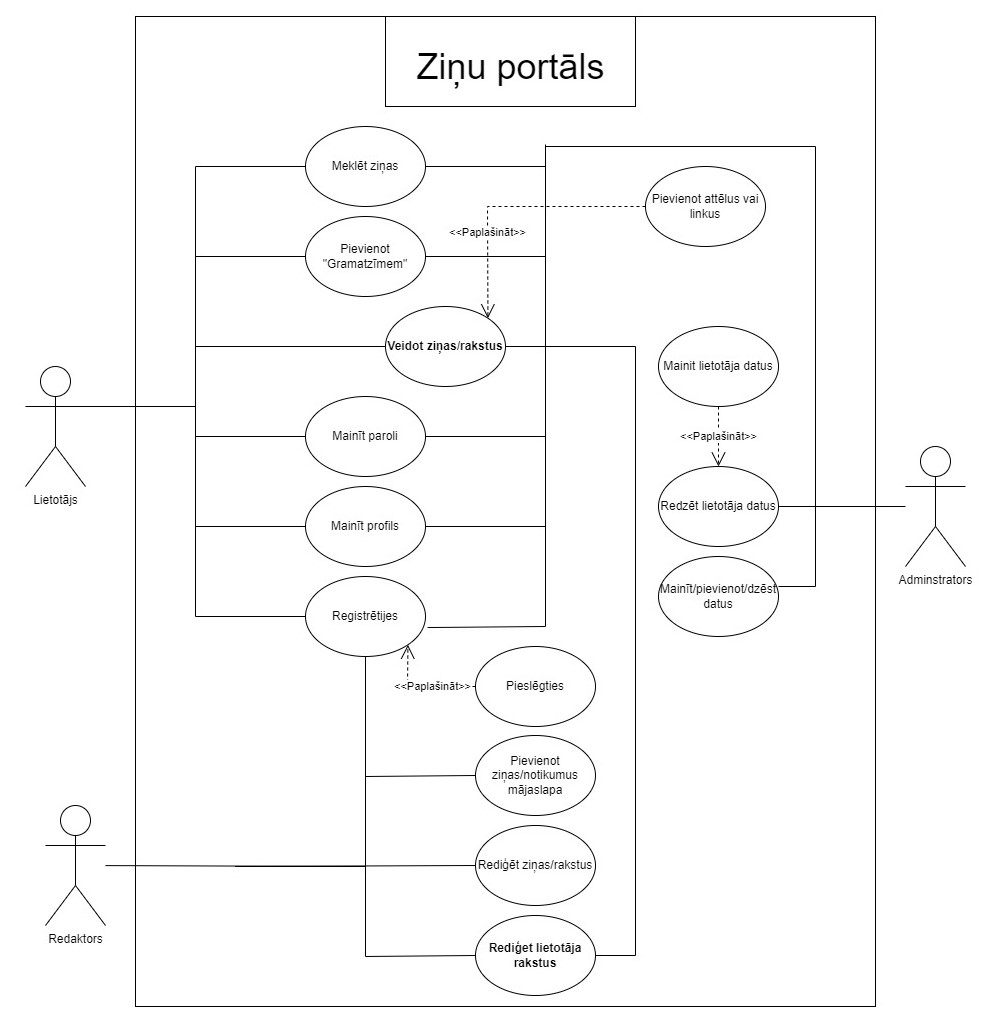
# UZDEVUMA NOSTĀDNE

Noslēguma darba uzdevums ir nodrošināt sabiedrībai uzticamu un patiesu ziņu un rakstu avotu, kas spēs piesaistīt auditorijas uzmanību. Darba saskarne būs paredzēta nepieredzējušiem lietotājiem, un tā būs viegli lietojama, lai nerastos kādas problēmas.

Izmantojot šo projektu, lietotāji var reģistrēties, lasīt jaunākās ziņas mūsu ziņu plūsmā un rakstīt savus rakstus vienā no šīs vietnes sadaļām par jebkuru no piedāvātajām tēmām, un pēc moderēšanas lietotāja raksts tiks publicēts un pieejams visiem pārējiem lietotājiem.

Šajam ziņu portālam ir jāizpilda sekojošās funkcionalitātes (skat. 1.1.att.):

* *iespēja pievienot ziņas plūsmai (pieejama administratoriem un redaktoriem);*
* *spēja rakstīt lietotāja rakstus;*
* *ziņu dzēšana, pievienošana, pārlūkošana, meklēšana un šķirošana;*
* *klientu pievienošana, dzēšana un rediģēšana;*
* *iespēja skatīt aktuālās ziņas tīmekļa vietnē;*
* *iespēja uzrakstīt atbalsta vēstuli;*
* *personīgo kontu rediģēšana.*



1.1.att. Lietojumgadījuma diagramma

# PRASĪBU SPECIFIKĀCIJA

## 2.1. Ieejas un izejas informācijas apraksts

### 2.1.1. Ieejas informācijas apraksts

Tā kā vietne ir ziņu un rakstu portāls, redaktoriem būs iespēja veidot ziņu ierakstus (ievadīt informāciju teksta un attēlu veidā), un lietotājiem veidot amatieru rakstus par dažādām tēmām (arī ievadīt tekstu un attēlus). Arī kādi dati ir nepieciešami lietotāja reģistrēšanai.

Zemāk ir tabulas no datu bāzes, no kuras ir iespējams saprast, kādi dati tiks ievadīti, kādi ierobežojumi pastāv (simbolu skaits un t.t.) un kādi dati ir nepieciešami.

Izmers nozimē maksimālo rakstzīmju numuru.

Sistēmā tiks nodrošināta šādas ieejas informācijas apstrāde.

1. Informācija par **Profilu** sastāvēs no šādiem datiem.

* Email – burtu teksts ar izmēru no 7 līdz 100 rakstzīmēm.
* Parole – burtu teksts ar izmēru līdz 255 rakstzīmēm.

2. Informācija par **Lomu (administrators vai redaktors vai lietotājs)** sastāvēs no šādiem datiem.

* Nosaukums – burtu teksts ar izmēru līdz 11 rakstzīmēm.

3. Informācija par **Ziņas** sastāvēs no šādiem datiem.

* ID – ciparu numurs ar izmēru līdz 100 rakstzīmēm.
* Virsraksts – burtu raksts ar ziņas virsrakstu.
* Kategorija – ziņas vai raksta kategorijas nosaukums līdz 1000 rakstzīmēm.
* Saturs – ziņas vai raksta būtiskais teksts līdz 10000 rakstzīmēm.
* Attēlas - jpg/jpeg/png formāta faili. (Pievienots katrai ziņai).

4. Informācija par **Kategoriju** sastāvēs no šādiem datiem. (Ievadīta no klaviatūras)

* Nosaukums – burtu teksts ar izmēru līdz 200 rakstzīmēm.

### 2.1.2. Izejas informācijas apraksts

1. **E-pasta paziņojums** lietotājiem pēc reģistrācijas - paldies par reģistrāciju.
2. **Web lāpa** ar kategoriju nosaukumiem(Izvelētas tēmas nosaukums).
3. **Web lāpa** ar cilvēka profilu informacīju.
4. Vietnē «Ziņu portāls» būs iespēja izmantot meklēšanas vaicājumu konkrētiem rakstiem, un meklēt pec: Nosaukumu; Saturu; Temai;.
5. Informācija tiks parādīta lietotāja ekrānā. Vietnē tiek rādīta tāda informācija kā: Izveidotas ziņas, lietotāju raksti un vietnes atjauninājumi. Administratoriem tiks parādīta papildu informācija, piemēram: Visi lietotāja dati, visa informācija par ziņām (ziņas un lietotāja rakstus), vietņu un lietotāju statistika (mērķauditorijas atlasei un auditorijas analīzei).

## 2.2. Funkcionālās prasības

Lai piekļūtu vietnes funkcionalitātei, lietotājam ir nepieciešams reģistrēties. Reģistrējoties lietotājam ir jāievada segvārds, e-pasts un parole. Katrs lauks ir nokārto validāciju. Tiek izveidots datu bāzes savienojums. Ja reģistrācija bija veiksmīga, dati tiek ierakstīti datu bāzē un lietotājam tiek izveidots profils.

Vietnē būs pieejamas šādas funkcijas:

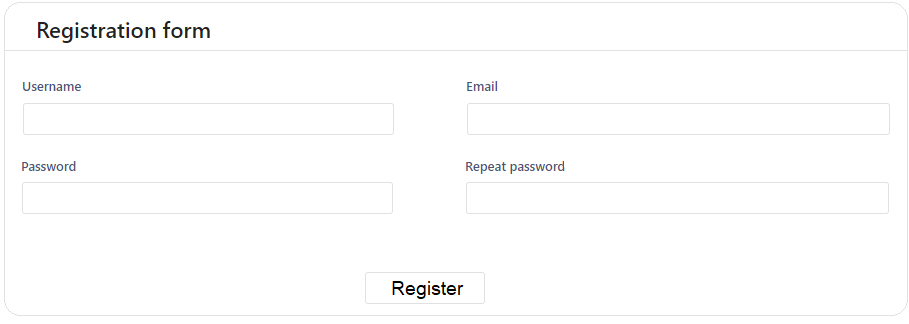
1. Jānodrošina iespēja reģistrēt jaunu lietotāju.
   1. Jāparedz ieejas informācijas par lietotāju (skat. 3.att.) ievadīšana un pārbaude un formāta pareizību.
   2. Ja kāds no obligātiem laukiem nav ievadīts, tad izvadīt par to kļūdas paziņojumu.
   3. Salīdzināt ievadīto lietotāja e-pasts ar sistēmā jau eksistējošo lietotāju e-pastiem un izvadīt paziņojumu, ja tie sakrīt.
2. Jānodrošina lietotāja autorizācija.
   1. Ja lietotājs eksiste un parole ir pareiza, sistēmai ir jānodrošina autorizāciju pieslēdzoties ar e-pastu un paroli.
   2. ja lietotājs ir bloķēts, nebūs iespējams pieteikties attiecīgajā profilā.
   3. Ja kāds no laukiem nav ievadīts, izvadīt par to paziņojumu.
   4. Ja lietotājs ir dzēsts, izvadīt par to paziņojumu.
3. Jānodrošina lietotājam veidot rakstus.
   1. Ziņu un lietotāju rakstu izveide;
   2. Ziņu un lietotāju rakstu lasīšana;
4. Administratoriem būs piekļuve unikālām ierobežotas piekļuves funkcijām, piemēram:
   1. Lietotāja pievienošana vai dzēšana sistēmā, izmantojot piekļuvi datu bāzei;
   2. Ziņu vai rakstu piespiedu pievienošana vai dzēšana bez nepieciešamības moderēt;
   3. Ja nepieciešams, dzēst lietotāju;
5. Jānodrošina administratoram veidot jaunas kategorijas
6. Jānodrošina lietotājiem apskatīt savu profilu.
7. Jānodrošina lietotājam mainit savo paroli.
8. Jānodrošina lietotājam pārskatīt kategorijas.

## 2.3. Nefunkcionalās prasības

1. Mājas lapai būs angļu valodā.
2. Pielāgojiet vietni dažādām ekrāna izšķirtspējām.
3. Ieviest aizsardzību pret DDoS uzbrukumiem (datu aizsardzība un vietņu aizsardzība).
4. Izveidojiet gaišu un tumšu vietnes tēmu.
5. Izveidojiet ērtu un vienkāršu logu ziņu veidošanai (modālais logs).
6. Izveidojiet vietni vienotā stilā, izmantojot trešo pušu paplašinājumus un resursus (piem. ‘Bootrstrap’), padarot vietni lietotājam draudzīgu.
7. Sistēma ir jābūt atjaunināmai, lai nodrošinātu jaunākās funkcijas un drošības.
8. Sistēmas kļūdas paziņojumi ir jābūt skaidriem un saprotamiem.
9. Sistēma ir jābūt pieejama 24 stundas diennaktī, 7 dienas nedēļā.

Sistēmas ekrānu skices:

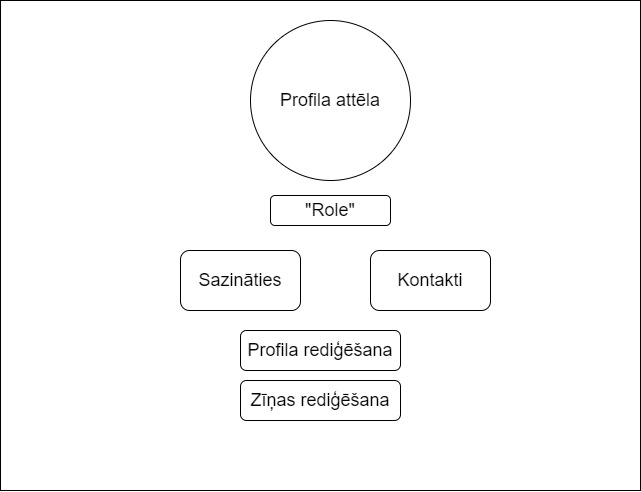
* Sistēmas reģistrācijas skice (skat 2.2. att.)



2.2.att. Sistēmas reģistrācijas skice

Šī skice attēlo sistēmas interfeisu, kurā lietotāji var reģistrēties. Skatā var redzēt pogu, kas ļauj piereģistrēt lietotāju.

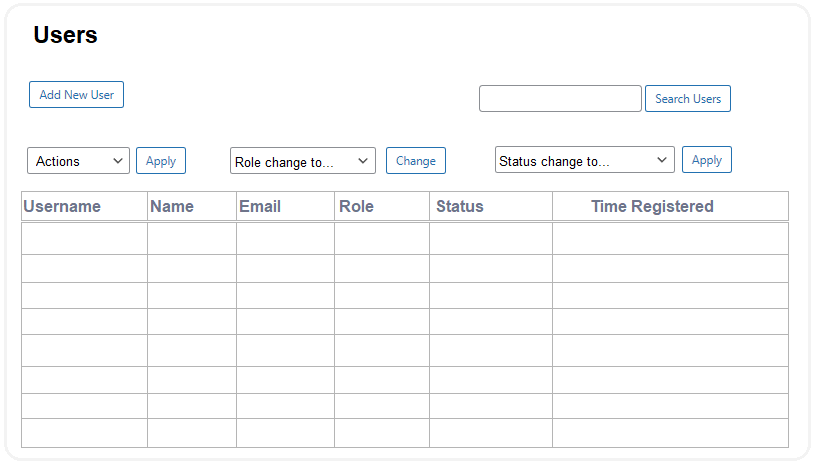
* Sistēmas profila skice (skat 2.1. att.)



2.1.att. Sistēmas profila skice

Šī skice attēlo profila interfeisu, ar lietotāju var sazināties, mainit profiku ar pogu “User update” un skatīt lietotāja ziņas postus ar pogu “News update”. Funkcionalitāte bus paplašināta nakotne.

* Sistēmas administratora paneļa skice (skat 2.3. att.)



* 2.3.att. Sistēmas administratora paneļa skice

Šī skice attēlo sistēmas interfeisu, kura administratori var apskatīt informāciju par lietotājiem, pievienot, dzēst vai rediģēt viņus.

* Ziņas izveidošanas formas skice (skat 2.4. att.)



2.4.att. Ziņas izveidošanas formas skice

# UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS

Sistēma ir dinamiska tīmekļa lapa. Vietnes izveidē tika izmantots **«Wordpress»** framevorks, jo tas ļāva sasniegt vēlamo rezultātu [7].

Dinamiska vietne ģenerē saturu servera pusē, reaģējot uz lietotāja mijiedarbību. Mūsu tīmekļa lappa izmanto datu bāzes un servera puses skriptu valodas PHP un MySQL, ar XAMPP palidzību lai radītu interaktīvāku un personalizētāku pieredzi. Ziņu portāls ir atkarīgs no dinamiski atjaunināta satura, lai lietotāji būtu informēti par jaunākajiem notikumiem. Tas ietver regulārus ziņas atjauninājumus un papildinājumus, nodrošinot, ka informācija ir aktuāla.

Ziņu portālam nebūs savas aplikācijas, kuru varēs lejupielādēt piemēram no "AppStore" vai "Play Market".

Sistēma ir paredzēta pārlūkprogrammai, tādēļ to realizācijai tiek izmantoti jaunākie programmatūras risinājumi.

Datubāzes istrādei tika izmantota **phpMyAdmin**, jo ir veidota vairākiem nolūkiem: datu tabulu un tabulu shēmu izveide, procedūru un funkciju rakstīšana. Lai sazinātos ar datubāzi, tiks izmantots arī phpMyAdmin, kas pados informāciju uz frontend daļu (HTML, CSS un PHP).

Datubāzes tika izmantots **MySQL**, versija MariaDB. Tika izmantota MySQL datubāze, jo ta atbalsta Microsoft SQL Sever Management Studio.

Lietotāja daļa tiek veidota, izmantojot dažas valodas, kas aprakstītas turpmāk:

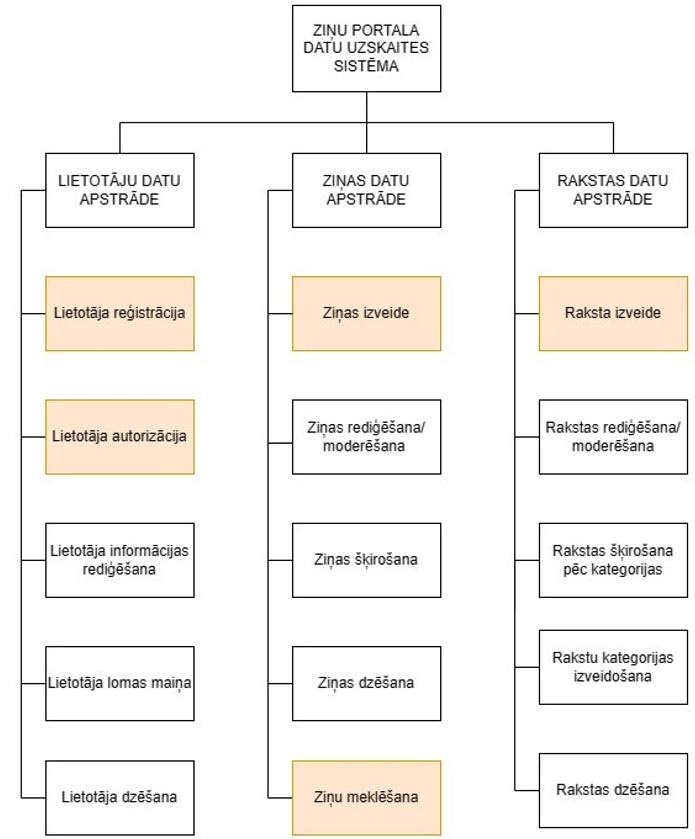
* **PHP -** Programmēšanas valoda, kas ļāva vietnei izskatīties tā, kā tā izskatās. Gandrīz viss vietnē darbojas ar PHP. Vispiemērotākā valoda veicamo uzdevumu veikšanai.
* **HTML** **-** izmantots tīmekļa lapu front-end daļas projektēšanai. Tā funkcionalitāte ir ideāli piemērota uzdevumam izveidot vietnes izskatu un sadalīt saturu blokos.
* **CSS -**  ir vienkārši izstrādāta valoda, kas paredzēta, lai vienkāršotu tīmekļa lapu izskatu veidošanas procesu. vietnē css ļauj izveidot lieliskas redzamā satura animācijas
* **JavaScript -** ir slavena skriptu valoda, ko izmanto, lai vietnēs radītu burvību un padarītu vietni lietotājam interaktīvu. To izmanto, lai rakstu un ziņu pievienošana un izskats būti vispar iespējams.
* **Microsoft Visual Studio Code 2022**, versija – 17.2.5. Tās ir jaudīgs ietvars, kas atbalsta PHP valodu un pietiekoši jaudīgs, lai veidotu aplikācijas un vietni.

# PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

## 4.1. Sistēmas struktūras modelis

### 4.1.1. Sistēmas arhitektūra

Sistēma būs 3 apakšsistēmās (skat. 4.1. att.), kas ir cieši savstarpēji saistītas. Katra no šīm apakšsistēmām atbild par konkrētu funkcionalitāti, nodrošinot integrētu pieeju sistēmas darbībai, ko uzskatāmi parāda funkcionālās dekompozīcijas diagramma.



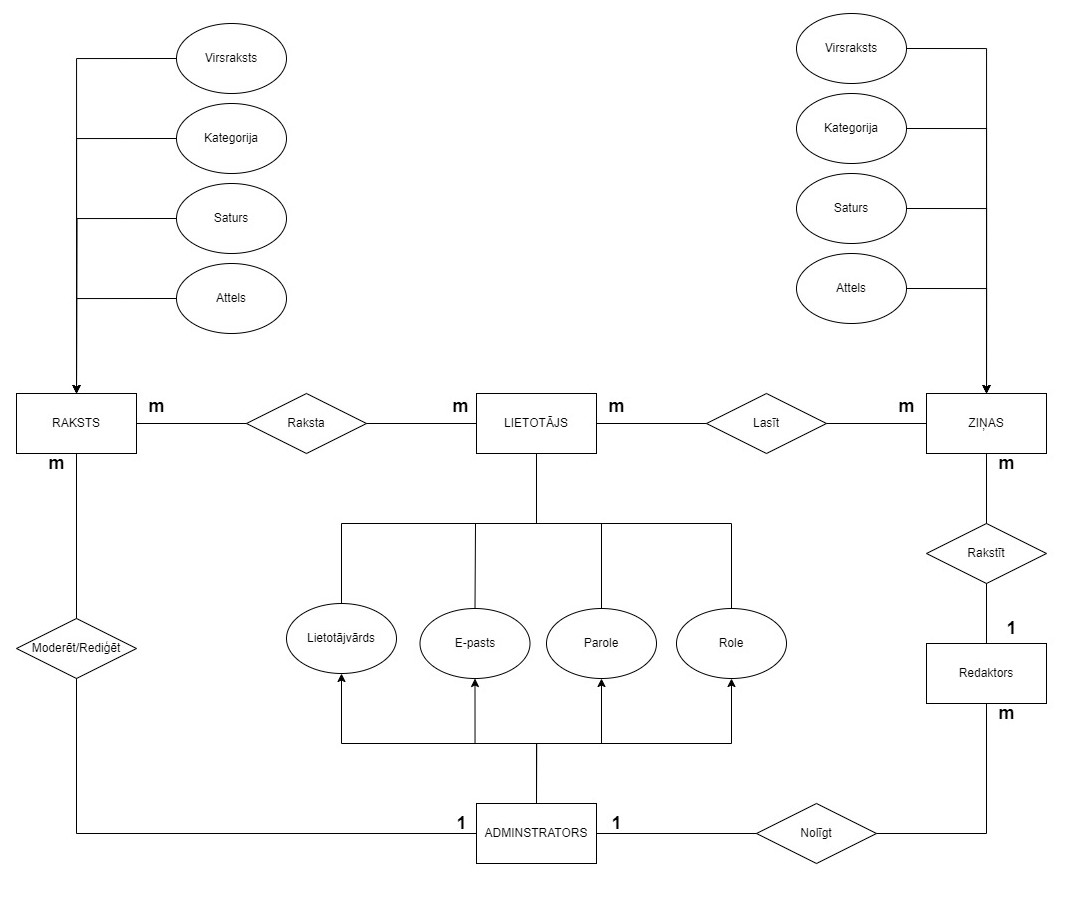
4.1.att. Funkcionālās dekompozīcijas diagramma

**Lietotāju datu apstrādes apakšsistēma -** ir atbildīga par visiem lietotāju datiem un to pārvaldību. Tas ietver lietotāja reģistrāciju, autorizāciju, iespēju rediģēt savus datus un visas saistītās informācijas glabāšanu. Apakšsistēma nodrošina, ka lietotāji var droši autentificēties, pārvaldīt savu konta informāciju, kā arī piekļūt visām tiem pieejamajām funkcijām, atkarībā no viņu tiesībām.

**Ziņas datu apstrādes apakšsistēma -** ir atbildīga par visu ziņas informācijas apstrādi un uzturēšanu. Tas ietver jaunas ziņas pievienošanu, ziņas dzēšanu, esošas ziņas datu maiņu (rediģēšanu) un dzēšanu, šķirošanu un meklēšanu pēc kategorijam, saturam vai virsrakstam. Ziņās komentēšana nebūs iespējams pēc būtības. Apakšsistēma nodrošina, ka ziņas ir aktuāli atspoguļots lietotājiem, kuri vēlas uzzināt kas notiek pasaulē.

**Rakstus datu apstrādes apakšsistēma -** ir atbildīga par visu saistīto ar rakstus informācijas apstrādi un pārvaldīšanu. Šī ir viena no galvenajām vietnes iezīmēm, ziņu un interesantu rakstu līdzāspastāvēšana vienuviet. Tas ietver rakstu izveide, to saturu maiņa (pēc nepiecešamības) vai dzēšana. Rakstiem tiks ieviesta iespēja atstāt komentārus un tos dzēst. Rakstiem būs vairākas iespējamās kategorijas, kuras lietotājs varēs izvēlēties, veidojot vienu. Jaunas kategorijas ir iespējams veidot. Apakšsistēma nodrošina, ka visi raksti ir vienmēr interesānti, ir patiesi, nevis dezinformācija.

### 4.1.2. Entītiju relāciju datu modelis (ER-diagramm)

Lai izveidotu datu bāzes shēmu vai datu modeli un izprastu veidojamās datu bāzes uzbūvi un pamatprincipus, tika izmantota ER diagramma (skat. 4.2. att.). Tas sastāv no entītijām un relācijām un ļauj datus uztvert kā objektus un izprast to savstarpējo saistību.

4.2.att. Sistēmas ER diagramma

* **Lietotājs** – pirmais lietotāju tips, var meklēt un skatīties ziņas.
* **Redaktors** – otrais lietotāju tips, var moderēt un rediģēt lietotājas rakstus un ziņas.
* **Administratos** – trešais lietotāju tips. Var darīt viss, kas pieējams citiem lietotājiem, bet arī moderēt pašus lietotājus (pirmo un otro lietotāju tipu). “Appoints” nozīme ka administrators var pievienot vai nolīgt redaktorus.
* **Ziņas** – Entitīja kas redaktors var izveidot un rediģēt.
* **Raksts** – Entitīja kas lietotājs var izveidot un papildināt, un redaktors var moderēt pirms publicēšānas.
* **Profils** – Entitīja kas piedēr katram lietotājām, un to var rediģēt pēc dizaina, un pēc lietotāja datiem kas iespējams tikai administratoram.

Manā Sistēmas ER diagramma starp klases objektiem ir attiecības viens pret daudziem vai daudzi pret daudziem.

* Viens pret daudziem attiecības nozīmē, ka vienam objektam var būt saistīti vairāki citi objekti, bet citam objektam var būt saistīts tikai viens objekts.
* Daudzi pret daudziem attiecības nozīmē, ka dažiem objektiem var būt saistīti vairāki citi objekti, un otrādi - citiem objektiem var būt saistīti arī vairāki citi objekti.

Manā diagrammā objekti ir savienoti šādi:

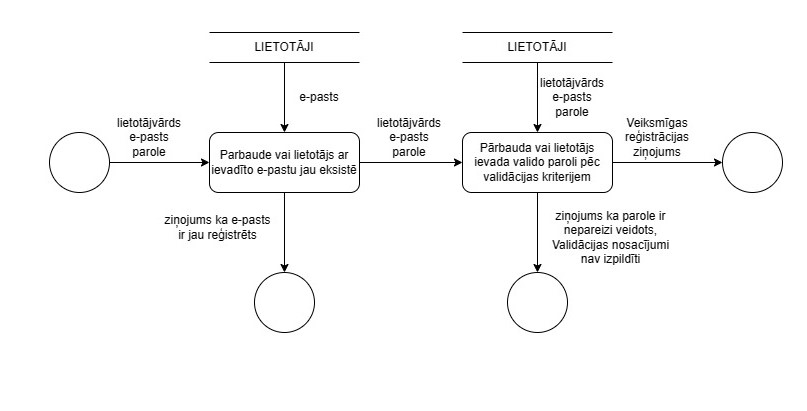
* Tabulā **“Lietotājs”** – attēloti individuāli sistēmas lietotāji, katram lietotājam ir savs sēgvards, e-pasts, parole un statuss. “Users” attiecas pret “Raksts” ka daudz prēt daudziem, jo daudzie lietotāji var rakstīt daudz rakstus. Arī attiecība starp “Lietotājs” un “Ziņas” ir daudz pret daudziem, jo daudzie lietotāji var lasīt daudz ziņas.
* Tabulā **“Ziņas”** – atsevišķa tabulā, tos var veidot “Redaktors” kas ir lietotāju tips. ”Ziņas” pret “Redaktors” attiecas kā “viens pret daudziem. Daudzas ziņas var veidot dažādi “Redaktori”. Daduzas ziņas var lasīt daudz lietotājus.
* Tabula **“Raksts”** – atsevišķa tabula. Rakstus var veidot “Lietotājs”. Viens lietotājs var veidot daudzus rakstus. Viens “Administrators” var moderēt dažādus rakstus. “Administrators” ir lietotāju tips.

## 4.2. Funkcionālais sistēmas modelis

### 4.2.1. Datu plūsmu modelis

**1**. **Lietotāja registrācija** (skat. 4.3. att.).

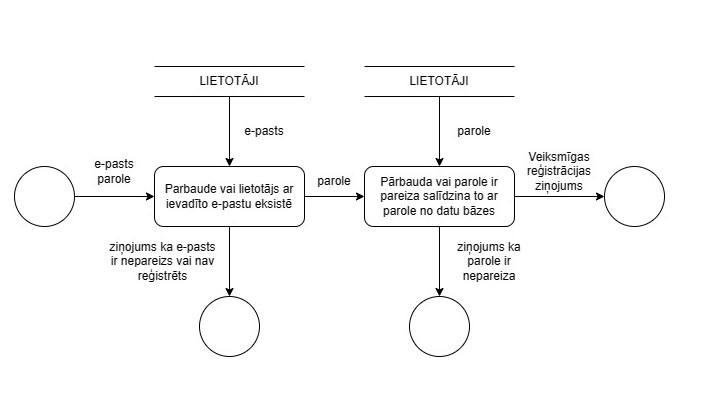
Pēc lietotāja datu ievades sistēma veic divas pārbaudes. Vispirms sistēma pārbauda, vai lietotājs ar tādu e-pastu jau eksistē, un tad pārbauda, vai lietotājs ir ievadījis pareizo (valido) parole pec validācijas kriterijem. Ja dati pārbaudīti veiksmīgi, tie tiek pievienoti datu bāzei, un sistēma informē lietotāju par veiksmīgu reģistrāciju. Ja kāda no pārbaudēm nav izpildīta, sistēma informē lietotāju par kadu kļūdu un izvada ziņu. [14]



4.3.att. Datu plūsmu diagramma – reģistrācija

1. **Lietotaja autorizācija** (skat. 4.4. att.).

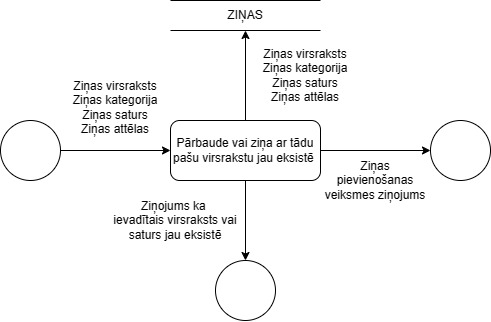
Pēc lietotāja datu ievades tie veic divas pārbaudes. Vispirms sistēma pārbauda, vai eksistē lietotājs ar tādu e-pastu, un vai lietotājs ir ievadījis pareizo paroli savam e-pastam. Ja dati ir veiksmīgi pārbaudīti, sistēma informē lietotāju par veiksmīgu autorizāciju. Ja kāda no pārbaudēm nav izpildīta, sistēma informē lietotāju par kļūdu un izvada ziņu. [14]



4.4.att. Datu plūsmu diagramma – autorizācija

1. **Ziņu izveidošanas process** (skat. 4.5. att.).

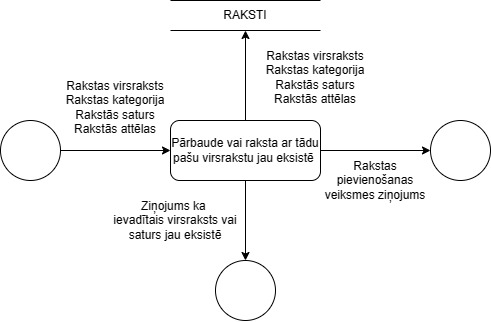
Pēc ziņas datu ievades tie veic vienu pārbaude. Sistēma pārbauda, vai ziņa ar tādu pašu virsrakstu vai saturu jau eksistē. Kategorijas varēs atkartotas dažadiem ziņam kā ari attēlas. Ja dati ir veiksmīgi pārbaudīti, sistēma informē lietotāju par veiksmīgu datu ievade un rakstu nosūtot moderēšanai. Ja administrators vai redaktors apstiprina zuņu, izmantojot "Wordpress" ietvara administratora paneli, ziņa tika publicēta. Ja kāda no pārbaudēm nav izpildīta, sistēma informē lietotāju par kļūdu un izvada ziņojumu. [12][13]



4.5.att. Datu plūsmu diagramma - ziņu izveidošanas process

1. **Rakstu izveidošanas process** (skat. 4.6. att.).

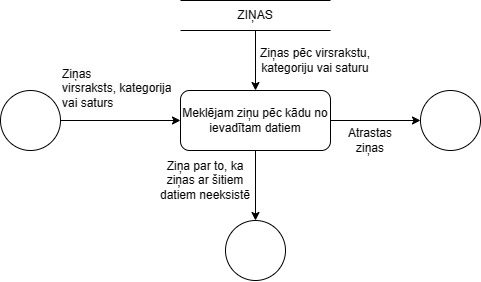
Rakstu veidošanas procesā viss ir tads pats ka ir ziņas veidošanai. Pēc raksta datu ievades tie veic vienu pārbaude. Sistēma pārbauda, vai raksts ar tādu pašu virsrakstu vai saturu jau eksistē. Kategorijas varēs atkartotas dažadiem rakstiem kā ari attēlas. Ja dati ir veiksmīgi pārbaudīti, sistēma informē lietotāju par veiksmīgu datu ievade un rakstu nosūtot moderēšanai. Ja administrators vai redaktors apstiprina rakstu, izmantojot "Wordpress" ietvara administratora paneli, raksts tiks publicēts. Ja kāda no pārbaudēm nav izpildīta, sistēma informē lietotāju par kļūdu un izvada ziņojumu. [12][13]



4.6.att. Datu plūsmu diagramma - rakstu izveidošanas process

1. **Ziņas vai rakstus meklēšanas process** (skat. 4.7. att.).

Pēc lietotaja kadus datus ievadīšanas meklēšanas joslā – sistēma mēkle ziņas ar tadus datus un izvada to. Ja tādu ziņu nav, sistēma izvāda ziņu ka ziņa neeksistē. [8]



4.7.att. Datu plūsmu diagramma – meklēšana

# 5. DATU STRUKTŪRAS APRAKSTS

## Datu bāzes fiziskā struktūra

Datu bāzes projektēšanas rezultātā tika veidotas vairākas tabulas un starp tām tika definēta tabulu relācija jeb saistība, kas norāda, ka abās tabulās saistītajos laukos ir vienādi dati.

Datubāze “ap-site” sastāv no sešpadsmit automātiski ģenerētam tabulām kuras satur informāciju par lietotājiem, kategorijam, ziņam un rakstam.

Vissvarīgākie tiks norādīti 1. līdz 6. tabulas, pārējās tām seko, jo lielākā daļa tabulu satur nesvarīgu informāciju vai tukšas. Kā minēts iepriekš, tā kā bija iesaistīts tīmekļa ietvars (framework) tas pats ģenerē tabulas. [8]

* Tabula **“wp\_users”** ar 10 laukiem. **–** glabās informāciju par lietotājus, kuri jau ir paspējuši reģistrēties sistēmā.
* Tabulas “wp\_users” struktūra

5.1 tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | ID | int | 20 | Lietotāja id numurs |
| 2 | user\_login | varchar | 60 | Lietotājvārds |
| 3 | user\_pass | varchar | 255 | Saglabā lietotāja “hash”paroli. |
| 4 | user\_nicename | varchar | 50 | URL draudzīgs lietotājvārds |
| 5 | user\_email | varchar | 100 | Lietotāja e-pasts |
| 6 | user\_url | varchar | 100 | Lietotāja URL |
| 7 | user\_registered | datetime |  | dat/laiks, kad lietotājs reģistrējās |
| 8 | user\_activation\_key | varchar | 255 | Vajag lai pieslēgties |
| 9 | user\_status | int | 11 | pārliecinieties, vai lietotājs ir reģistrēts |
| 10 | display\_name | varchar | 250 | Lietotāja lietotājvārds vietnē |

Primārā atslēga (PK) ir kolonna “ID”.

* Tabula **“wp\_terms”** ar 4 laukiem **–** glabās informāciju par pieejamās kategorijas, veidojot ziņas/rakstus.

Tabulas “wp\_terms” struktūra

5.2. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | term\_id | bigint | 20 | Kategorijas id |
| 2 | name | varchar | 200 | Kategorijas nosaukums |
| 3 | slug | varchar | 200 | Draudzīgs skaidrojošs nosaukums |
| 4 | term\_group | bigint | 10 | ļauj grupēt taksonomijas ietvaros |

Primārā atslēga (PK) ir kolonna “term\_id”.

* Tabula **“wp\_posts”** ar 18 laukiem **–** glabās informāciju par ziņas/rakstus, kas ir izveidoti.

Tabulas “wp\_posts” struktūra

5.3. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | ID | bigint | 20 | raksta id numurs |
| 2 | post\_author | bigint | 60 | Raksta autors |
| 3 | post\_date | datetime | - | Raksta izveidošanas datums |
| 4 | post\_content | longtext | - | Raksta neierobežots teksts |
| 5 | post\_title | text | - | Raksta neierobežots virsraksts |
| 6 | post\_status | varchar | 20 | Publicēts vai gaida pārskatīšanu |
| 7 | comment\_status | varchar | 20 | Iespēja veidot komentarus |
| 8 | post\_name | varchar | 200 | URL-draudzīgs raksta nosaukums |
| 9 | post\_modified | datetime | - | Laiks kad raksts tiek mainīts |
| 10 | post\_modified\_gmt | datetime | - | Laiks kad raksts tiek mainīts gmt |
| 11 | post\_content\_filtered | longtext | - | Ziņas satura filtrētā versija |
| 12 | post\_parent | longtext | - | Hierarhiskās attiecības starp rakst. |
| 13 | guid | varchar | 255 | Globāli unikāls identifikators |
| 14 | menu\_order | int | 11 | Norāda secību kad jāparādās rakst |
| 16 | post\_type | varchar | 20 | Klasificē saturu dažādos veidos |
| 17 | post\_mime\_type | varchar | 100 | Augšupielādēto pielikumu MIME veids |
| 18 | comment\_count | int | 20 | Komentaru skaits |

Primārā atslēga (PK) ir kolonna “ID”.

* Tabula **“wp\_usermeta”** ar 4 laukiem **–** programmā WordPress tiek izmantota, lai saglabātu metadatus par lietotājiem. Šī tabula ļauj saglabāt papildu informāciju par lietotājiem, kas nav iekļauta noklusējuma lietotāju tabulā (wp\_users). Katra lietotāja metadatu daļa tiek identificēta ar meta\_key, un tai ir saistīta meta\_vērtība.

Tabulas “wp\_usermeta” struktūra

5.4. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | umeta\_id | bigint | 20 | unikāls identifikators katrai lietotāja metadatu rindai |
| 2 | user\_id | bigint | 20 | lietotāja ID, kuram pieder šie metadati |
| 3 | meta\_key | varchar | 255 | metadatu atslēga (vai nosaukums) |
| 4 | meta\_value | longtext | - | metadatu vērtība |

Primārā atslēga (PK) ir kolonna “umeta\_id”.

* Tabula **“wp\_postmeta”** ar 4 laukiem **–** metadati par rakstiem un ziņam.

Tabulas “wp\_postmeta” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | meta\_id | bigint | 20 | unikāls identifikators katrai rakstai |
| 2 | post\_id | bigint | 20 | raksta ID, kurai pieder šie metadati |
| 3 | meta\_key | varchar | 255 | metadatu atslēga (vai nosaukums) |
| 4 | meta\_value | longtext | - | metadatu vērtība |

Primārā atslēga (PK) ir kolonna “meta\_id”.

* Tabula **“wp\_alm”** ar 5 laukiem **–** izmanto, lai vietnē izveidotu bezgalīgu ritināšanu un lappušu šķirošanu ziņu sarakstiem. framevorkā WP\_ALM jeb Ajax Load More ļauj ielādēt papildu ziņas, nepārlādējot lapu, kas uzlabo lietotāja pieredzi un var palielināt lietotāju iesaisti vietnē.

Tabulas “wp\_alm” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | id | mediumint | 9 | veidnes id |
| 2 | name | text | - | veidnes nosaukums |
| 3 | repeaterDefault | longtext | - | noklusējuma atkārtotāja veidnes iestatījumi |
| 4 | repeaterType | text | - | nosaka atkārtotāja tipu |
| 5 | pluginVersion | text | - | spraudņa versija |

Primārā atslēga (PK) ir kolonna “id”.

* Tabula **“wp\_commentmeta”** ar 4 laukiem **–** Šī ir WordPress datu bāzes tabula, kurā tiek glabāti papildu metadati komentāriem vietnē.

Tabulas “wp\_commentmeta” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | meta\_id | bigint | 20 | metadatu id |
| 2 | comment\_id | bigint | 20 | Komentāru id |
| 3 | meta\_key | varchar | 255 | Metadatu atslēga (vai nosaukums) |
| 4 | meta\_value | longtext | - | metadatu vērtība |

Primārā atslēga (PK) ir kolonna “meta\_id”.

* Tabula **“wp\_comments”** ar 15 laukiem **–** Šī ir WordPress datu bāzes tabula, kurā tiek glabāti visi jūsu vietnes lietotāju atstātie komentāri. Katrs ieraksts šajā tabulā ir viens lietotāja atstāts komentārs. Komentāru funkcija nav realizēta, bet tabula paliek proektā.

Tabulas “wp\_comments” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | comment\_ID | bigint | 20 | metadatu id |
| 2 | comment\_post\_ID | bigint | 20 | komentāru id |
| 3 | comment\_author | tinytext | 255 | komentāra autora vārds |
| 4 | comment\_author\_email | varchar | 100 | komentāra autora e-pasts |
| 5 | comment\_author\_url | varchar | 200 | komentāra autora URL |
| 6 | comment\_author\_IP | varchar | 100 | komentāra autora IP adrese |
| 7 | comment\_date | datetime | - | komentāra atstāšanas datums un laiks |
| 8 | comment\_date\_gmt | datetime | - | kad komentārs tika atstāts GMT formātā |
| 9 | comment\_content | text | - | komentara teksts |
| 10 | comment\_karma | int | 11 | komentāru vērtējums |
| 11 | comment\_approved | varchar | 20 | komentāra statuss |
| 12 | comment\_agent | varchar | 255 | autora izmantotais pārlūkprogrammas |
| 13 | comment\_type | varchar | 20 | komentara tips |
| 14 | comment\_parent | bigint | 20 | vecāku komentāra ID |
| 15 | user\_id | bigint | 20 | autora id |

Primārā atslēga (PK) ir kolonna “comment\_ID”.

* Tabula **“wp\_duplicator\_packages”** ar 7 laukiem **–** tabula, ko izmanto spraudnis Duplicator, lai pārvaldītu un saglabātu informāciju par izveidotajām pakotnēm. Duplicator spraudnis nodrošina rīkus vietņu dublējumu izveidei.

Tabulas “wp\_duplicator\_packages” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | id | bigint | 20 | 'iepakojuma' id |
| 2 | name | bigint | 250 | 'iepakojuma' nosaukums |
| 3 | hash | varchar | 50 | 'iepakojuma' heš-kods |
| 4 | created | datetime | - | kad ir izveidots |
| 5 | owner | varchar | 60 | kurš ir izveidojis |
| 6 | status | int | 11 | kāds statuss |
| 7 | package | longtext | - | 'iepakojums' |

Primārā atslēga (PK) ir kolonna “id”.

* Tabula **“wp\_fpsm\_forms”** ar 7 laukiem **–** tabula, kas ir saistīta ar spraudni Formidable Forms. Formidable Forms - spraudnis veidlapu izveidei un pārvaldībai vietnē.

Tabulas “wp\_fpsm\_forms” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | form\_id | mediumint | 9 | formas id |
| 2 | form\_title | varchar | 255 | formas virsraksts |
| 3 | form\_alias | varchar | 255 | nodrošina "pseidonimu" vai "alias" |
| 4 | form\_details | longtext | - | formas detaļas |
| 5 | post\_type | varchar | 255 | raksta tips |
| 6 | form\_type | varchar | 255 | formas tips |
| 7 | form\_status | mediumint | 9 | formas statuss |

Primārā atslēga (PK) ir kolonna “form\_id”.

* Tabula **“wp\_options”** ar 4 laukiem **–** tabula, kas ir saistīta ar spraudni Formidable Forms. Formidable Forms - spraudnis veidlapu izveidei un pārvaldībai vietnē.

Tabulas “wp\_options” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | option\_id | bigint | 20 | opcijas id |
| 2 | option\_name | varchar | 191 | opcijas nosaukums |
| 3 | option\_value | longtext | - | Opcijas vērtība, kas satur datus vai parametrus |
| 4 | autoload | varchar | 20 | Karogs, kas norāda, vai opcijai ir jāielādējas automātiski, kad vietne tiek ielādēta |

Primārā atslēga (PK) ir kolonna “option\_id”.

* Tabula **“wp\_termmeta”** ar 4 laukiem **–** tabula, kas tiek izmantota, lai saglabātu papildu metadatus taksonomijām (terminiem). Šī tabula ļauj saistīt patvaļīgus datus ar terminiem (kategorijām, tagiem un citām taksonomijām) sistēmā.

Tabulas “wp\_termmeta” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | meta\_id | bigint | 20 | metadatu id |
| 2 | term\_id | bigint | 20 | termina (taksonomijas) ID |
| 3 | meta\_key | varchar | 255 | metadatu atslēga |
| 4 | meta\_value | longtext | - | metadatu vērtība kas saistīta ar metadatu atslēgu |

Primārā atslēga (PK) ir kolonna “meta\_id”.

* Tabula **“wp\_term\_relationships”** ar 3 laukiem **–** tabula, kas saista ziņas ar taksonomijas terminiem, piemēram, kategorijām vai tagiem. Šī tabula attēlo attiecību kopu starp ierakstiem un tiem atbilstošajām taksonomijām.

Tabulas “wp\_term\_relationships” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | object\_id | bigint | 20 | objektu id |
| 2 | term\_taxonomy\_id | bigint | 20 | taksonomijas id |
| 3 | term\_order | int | 11 | objekta terminu kārtošanas secība |

Primārā atslēga (PK) ir kolonna “object\_id” un “term\_taxonomy\_id”.

* Tabula **“wp\_term\_taxonomy”** ar 6 laukiem **–** Šī ir tabula, kurā ir informācija par taksonomijām un to terminiem. Tam ir galvenā loma vietnes satura organizēšanā un klasificēšanā. [16]

Tabulas “wp\_term\_taxonomy” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | term\_taxonomy\_id | bigint | 20 | taksonomijas id |
| 2 | term\_id | bigint | 20 | termina id, kas ir saistīts ar šo taksonomiju. |
| 3 | taxonomy | varchar | 32 | taksonomijas nosaukums |
| 4 | description | longtext | - | taksonomijas apraksts |
| 5 | parent | bigint | 20 | vecāku taksonomijas id |
| 6 | count | bigint | 20 | taksonomiju saistīto terminu skaits. |

Primārā atslēga (PK) ir kolonna “term\_taxonomy\_id”.

* Tabula **“wp\_user\_registration\_sessions”** ar 4 laukiem **–** tabula, kas ir saistīta ar lietotāja reģistrācijas spraudni. Lietotāju reģistrācijas spraudnis nodrošina iespēju izveidot pielāgotas veidlapas lietotāju reģistrēšanai vietnē.

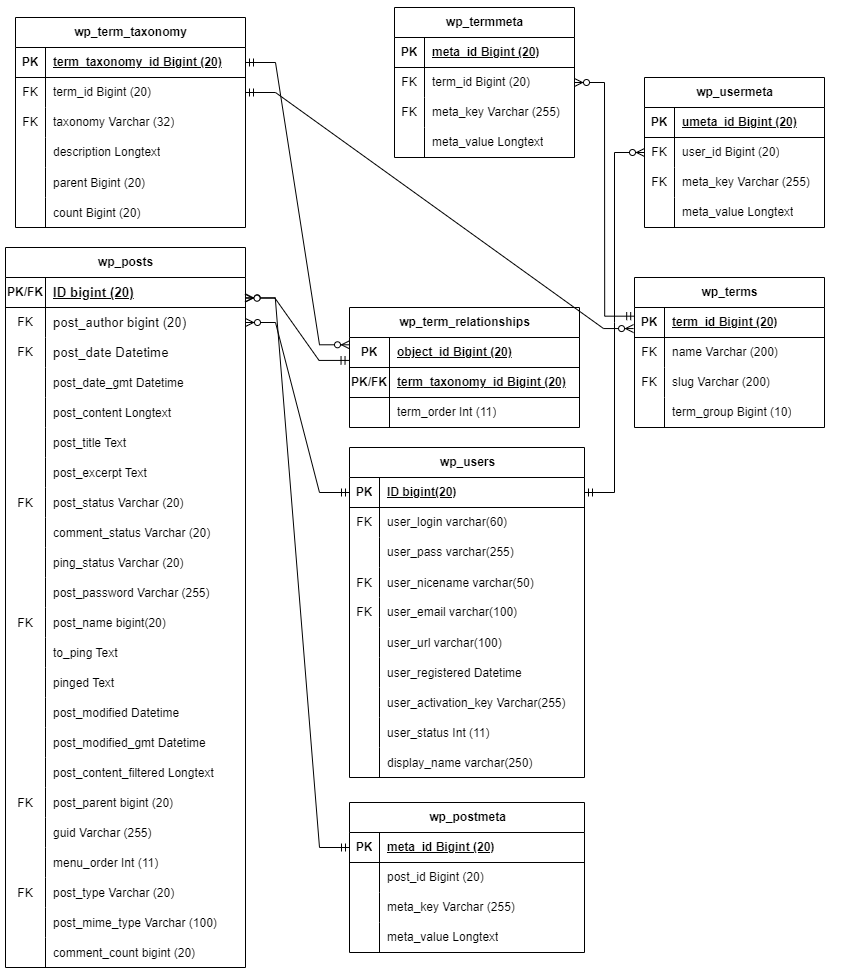
Tabulas “wp\_user\_registration\_sessions” struktūra

* 1. tabula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numurs | Lauka nosaukums | Datu tips | Izmērs | Apraksts |
| 1 | session\_id | bigint | 20 | sessijas id |
| 2 | session\_key | char | 32 | sessijas atslega |
| 3 | session\_value | longtext | - | vertiba |
| 4 | session\_expiry | bigint | 20 | kad sesija beidzas |

Primārā atslēga (PK) ir kolonna “session\_key”.

Neizmantotās tabulas netika ievietotas datu bāzes tabulu saišu shēma.

****

5.1.att. Datu bāzes tabulu saišu shēma

# 6. LIETOTĀJA CEĻVEDIS

## 6.1. Sistēmas prasības aparatūrai un programmatūrai

Lai sistēma darbotos pareizi, jums būs nepieciešams:

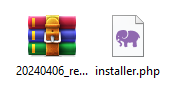
Aparatūras prasības:

* Operatīvā atmiņa: Vismaz 2 GB RAM.
* Interneta pieslēgums: Stabils interneta pieslēgums ar ātrumu vismaz 5 Mb/s
* Tīmekļa pārlūkprogramma: ieteicams izmantot pārlūkprogrammus Google Chrome, Mozilla Firefox, Safari vai Microsoft Edge
* Ekrāna izšķirtspēja: 1280x720
* JavaScript: pārlūkprogrammas iestatījumos ir iespējots JavaScript atbalsts, lai tīmekļa portālā pareizi darbotos interaktīvie elementi.
* Operētājsistēma: Windows 10 vai jaunāka versija / macOS 10.12 vai Linux 5.2.

## 6.2. Sistēmas instalācija un palaišana

Lai instalētu un izvietotu vietni, ir nepieciešams:

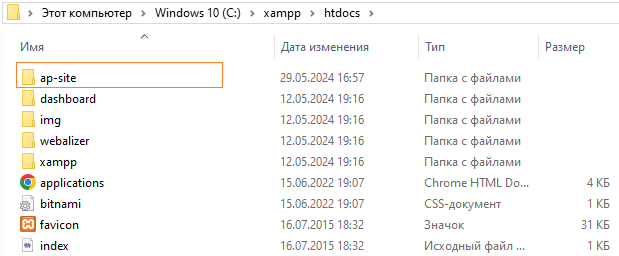
1. Ieviesiet vai instalēt datorā lietojumprogrammu, kas ļauj darbināt pilnvērtīgu tīmekļa serveri — XAMPP
2. Lejupielādējiet vietnes avota koda arhīvu ar nosaukumu ‘20240406\_retell\_d65d465db748501c7455\_20240531074306\_archive’ (nosaukums atkarīgs no vietnes versijas, pašreizējais nosaukums ir norādīts iepriekš).
3. Lejupielādējiet failu ‘installer.php’ (nāk kopā ar vietnes arhīvu), kas nepieciešams vietnes izvietošanai. [10]



6.1. att. Rezultāts pēc arhīva un uzstādītāju lejupielādēšanai

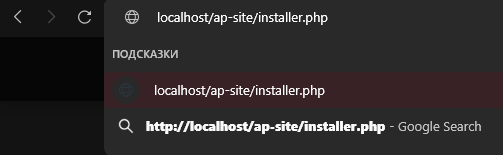
1. Lokāla XAMPP servera 'htdocs' mapē izveidojiet mapi ar jebkuru nosaukumu (manā gadījumā ap-site). Ceļš izskatās šādi: ‘C:\xampp\htdocs\’
2. Jāievieto arhīvu neizpakojot mapē “ap-site”.
3. Ievietojiet failu 'installer.php' ‘htdocs’ mapē pa ceļu ‘C:\xampp\htdocs\ap-site’

Tagad vietne atrodas pa ceļu ‘C:\xampp\htdocs\ap-site’ ar instalēšanas failu.



6.2. att. Rezultāts pēc arhīva izpakošanas

Tagad jebkurā pārlūkprogrammā meklēšanas laukā jāieraksta - localhost/(mapes-nosaukums)/installer.php (skat. 6.3. att.).

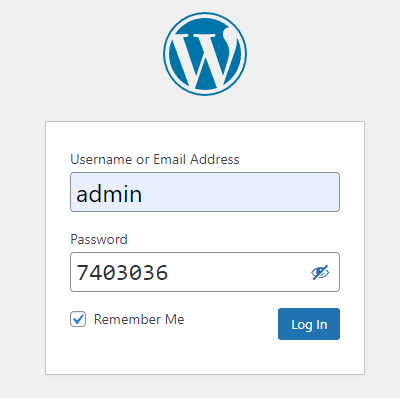


6.3. att. Pārlūkprogrammā pieprasījums, lai pārietu uz vietnes instalēšanu

1. Atvērtajā logā izvēlieties datu bāzes variantu ‘Create new database’, tālāk jāievada datu bāzes nosaukumu un lietotājvārdu, parole nav nepieciešama. Nospiežiet pogu ‘Next’.
2. Pēdējā instalēšanas posmā, atverot administratora paneli ‘Wordpress’, jāievada pieteikšanas vārds un parole. (skat. 6.4. att.).

Pieteikšanas vārd – admin

Parole – 7403036



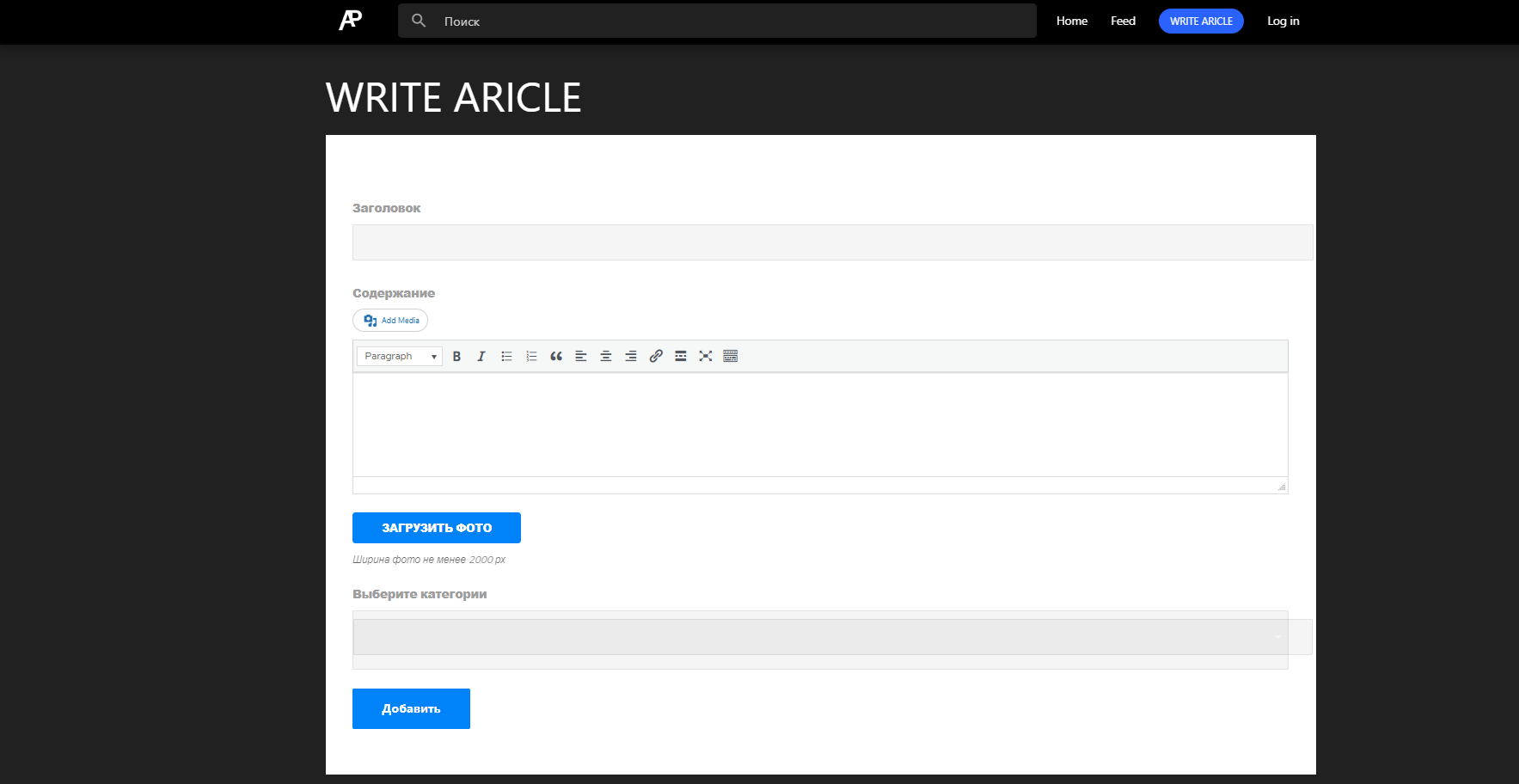
6.4. att. Piesakties WordPress kontā tīmekļa lapai

Tagad vietne ir veiksmīgi instalēta un izvietota jūsu datorā vietējā serverī. [10]

## 6.3. Programmas apraksts

* Atveriet savu pārlūkprogrammu:
* Datorā vai viedtālrunī palaidiet tīmekļa pārlūkprogrammu.
* Pārlūkprogrammas adreses joslā ievadiet ziņu portāla adresi. Šī adrese var tikt sniegta kā URL (piemēram, http://www.newsportal.com).
* Mājas lapa: Kad sākumlapa tiks ielādēta, jūs redzēsiet jaunākās ziņas, dažādas ziņu sadaļas un cilni, kas aizvedīs uz vietnes sadaļu ar dažādiem rakstiem. Galvenajā lapā parasti ir arī galvenās sadaļas un svarīgi jaunumi.
* Sadaļas navigācija: Izmantojiet lapas augšdaļā esošo navigācijas izvēlni, lai pārietu uz konkrētām ziņu sadaļām, piemēram, "Home" - galvena lapa, "Feed" – rakstu un ziņas plusma, "WRITE ARTICLE" – uzrakstit savu rakstu, ja esi reģistrēts, “Search” – meklēšanas josla un citas.
* Lasit ziņas/rakstus: Noklikšķiniet uz ziņu lauka "READ", lai lasītu pilnu tekstu. Ziņu lapā varat atrast arī attēlus, videoklipus un citus multivides elementus.
* Meklēt: Ja jums ir konkrēts pieprasījums, izmantojiet meklēšanas funkciju, lai ātri atrastu jūs interesējošo informāciju. Meklēt pēc virsraksta vai kategorijai.

### Rakstu/ziņu veidošana

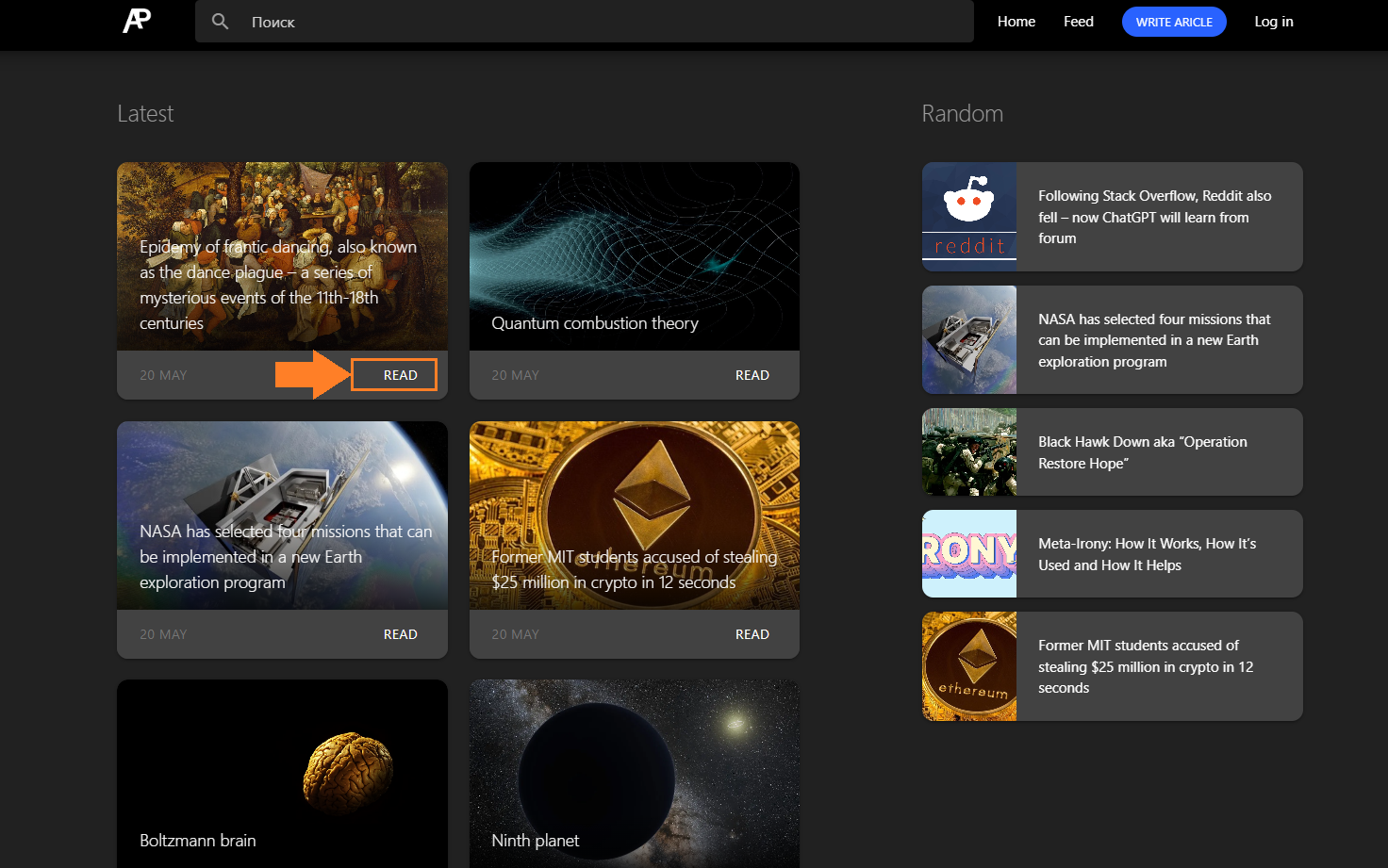
Raksta vai ziņas izveide, izmantojot piedāvāto formu (skat. 6.5. att.).

6.5.att. Piedavāta forma

Šajā ekrānuzņēmumā var redzēt kādiem parametriem ir jābūt rakstam, lai to publicētu. Tas ir: Virsraksts; Teksts, izvelēta kategorija un bilde ziņai vai rakstam.

### Rakstu/ziņu lasīšana

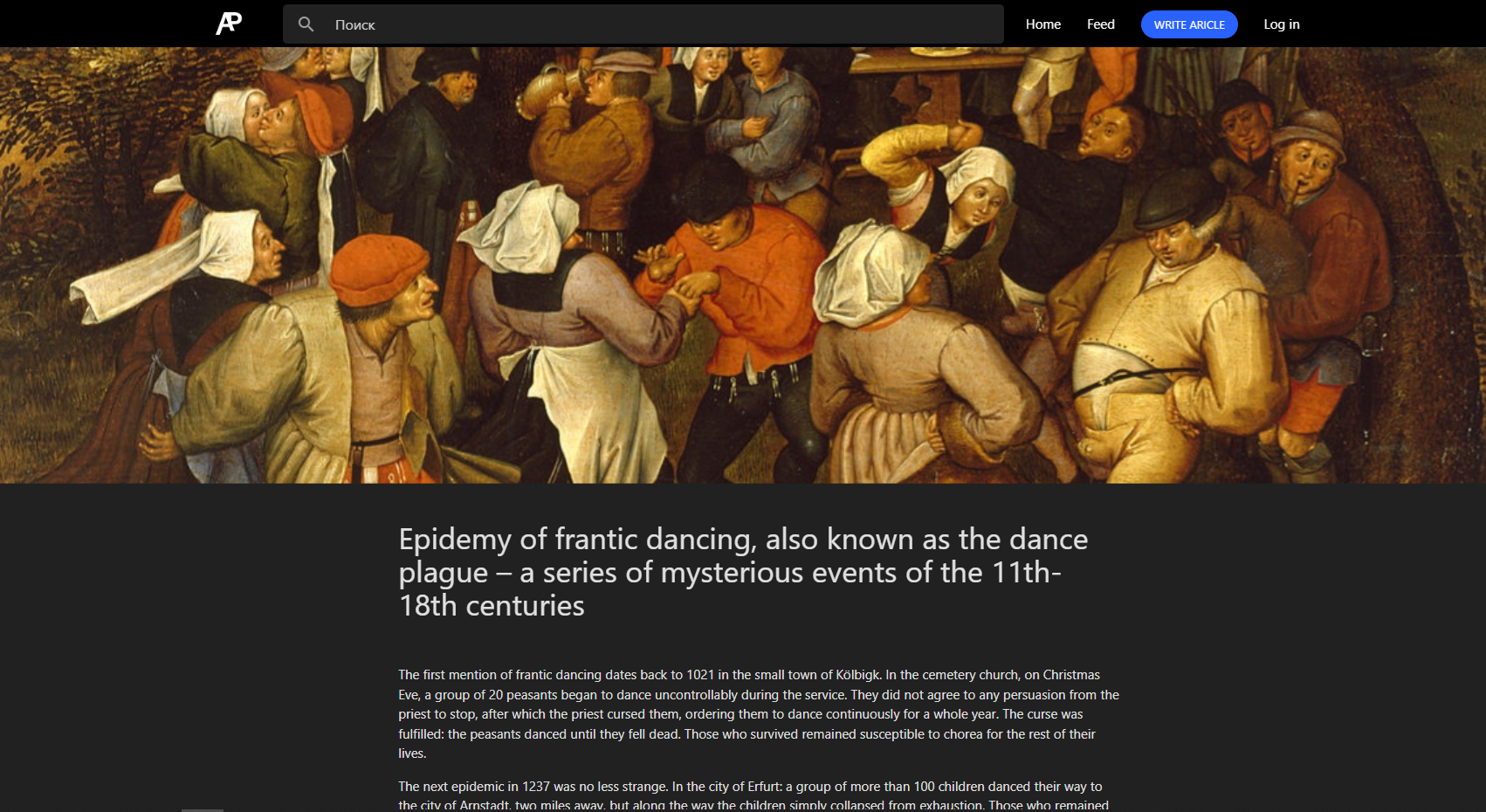
Raksta vai ziņas redzēšana un lasīšana galvenajā lapa (skat. 6.6. att.)

****

6.6.att. Galvena lapa – Mājas lapa

Šajā ekrānuzņēmumā ir redzama vietnes galvenā lapa, kurā varat doties uz jebkuru pieejamo rakstu un iepazīties ar to.

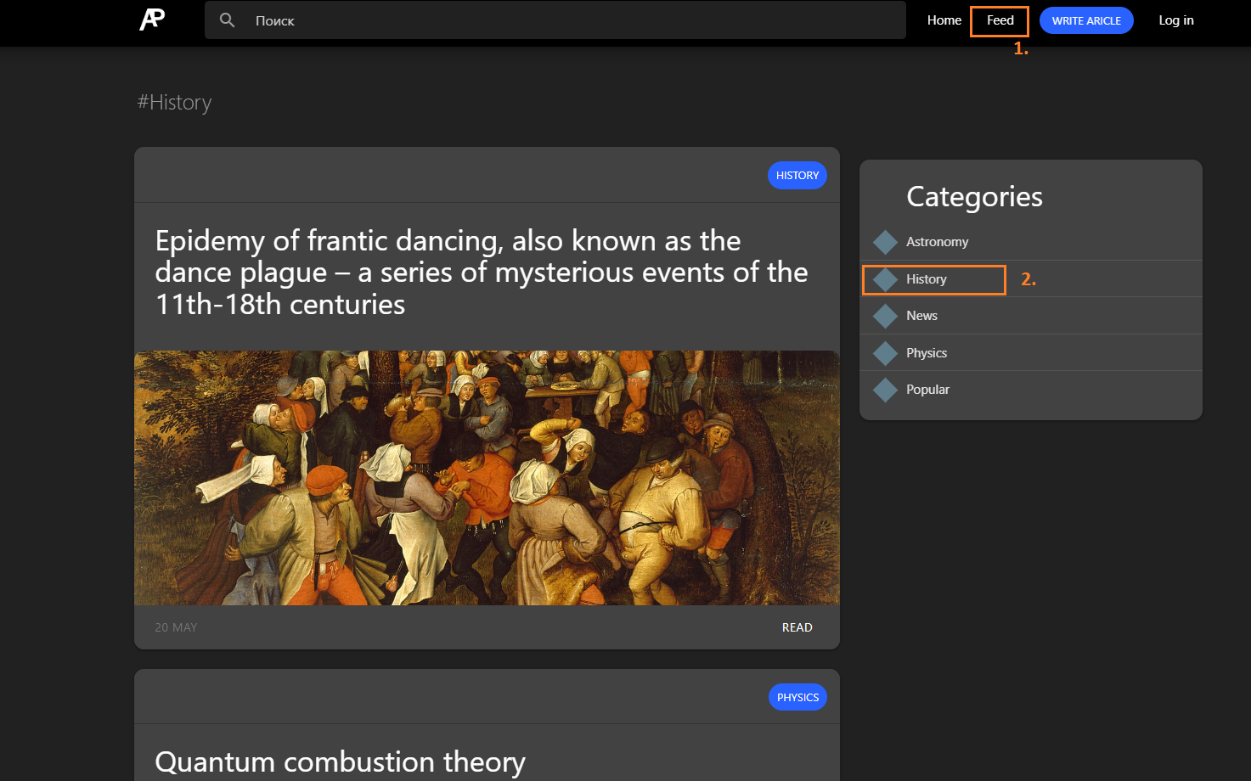
Šeit ir redzams, ka, noklikšķinot uz pogas “READ”, raksts tiek atvērts. Augšpusē ir redzama bilde, tad seko norādītais virsraksts, pēc tam raksta teksts. Vietnes pašā apakšā, tāpat kā jebkurā citā lapā, “footer” – kājene. (skat. 6.7. att.)



6.7.att. Pirmais atvērtais raksts

### Rakstu/ziņu kārtošana pēc kategorijas

Raksta vai ziņas redzēšana un lasīšana rakstu plusmā (skat. 6.8. att.)

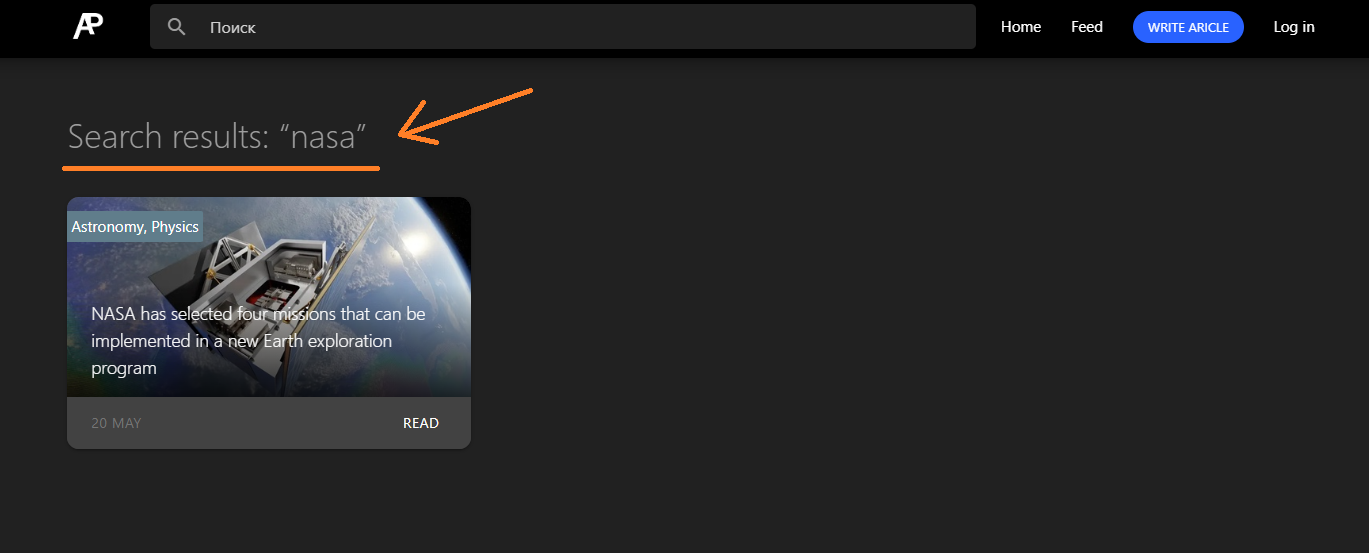


6.8.att. Rakstu plusmā iespēja kartot pēc kategorijam

Dodoties uz rakstu plūsmas lapu (poga ‘Feed’), mums ir iespēja izvēlēties kategoriju, kas mūs interesē, un izlasīt visus rakstus attiecīgajā kategorijā.

### Rakstu/ziņu meklēšana meklēšanas joslā

Raksta vai ziņas meklēšana meklēšanas joslā (skat. 6.9. att.)



6.9.att. Rakstu plusmā iespēja meklēt

Lapā ir iespēja meklēt interesējošo rakstu vai ziņas, izmantojot meklēšanas joslu. Ievadot tekstu meklēšanas laukā, rezultāts tiks atgriezts ar atbilstošo virsrakstu vai vārdu no raksta tekstā.

## 6.5. Testa piemērs

* 1. tabula

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Prasības numurs | Ievaddati/situācijas apraksts | Ievaddati/situācijas apraksts | Sagaidāmais rezultāts | Statuss |
| 1. | 1. | Iespēja reģistrēt jaunu lietotāju | Ievades dati: lietotājvards, e-pasts, parole un paroles atkartojums | Tiks pārbaudīta lietotāja datus atbilstība, vai lietotājvārds nav aizņemts, parole sakrīt ar atkartotu parole un e-pasts atbīlst validācijas nosacījumiem | Pareizi |
| 2. | Lietotāja e-pasts = platon.martynov20^gmail.com | Paziņojums ka e-pasts neatbilst validācijas kritērijiem | Pareizi |
| 3. | Nepareiza paroles atkārtota ievadīšana | Paziņojums ka parole neatbilst iepriekš ievadītam | Pareizi |
| 4. | 2. | Esoša lietotāja autorizācija | Pareizi dati: e-pasts, parole | Tiks pārbaudīta lietotāja esamība un tad lietotājs varēs izmantot vietnes funkcijas | Pareizi |
| 5. | Lietotāja e-pasts = platon.martynov@gmail.com | Paziņojums, ka e-pasts neatbilst nevienam kontam | Pareizi |
| 6. | Nepareiza parole | Paziņojums, ka daži dati ir ievadīti nepareizi | Pareizi |
| 7. | 3. | Jānodrošina lietotājam veidot rakstus | Ievadot datus laukos: virsraksts; saturs;, izvēloties piemēroto no piedāvātajām kategorijām un izvēloties attēlu ierakstam | Raksts ar visiem norādītajiem datiem tiek nosūtīts moderēšanai turpmākai publicēšanai lapā | Pareizi |
| 8. | Nav izvelota neviena bilde | Paziņojums, ka bildes izvēle ir nepieciešama, lai nosūtītu ziņu moderēšanai | Pareizi |
| 9. | Nav norādīts teksts, nosaukums vai kategorijas | Paziņojums, ka veidlapas iesniegšana ir veiksmīga | Nepareizi |
| 10. | 5. | Jānodrošina administratoram veidot jaunas kategorijas | Admin panelī sadaļā "Post" sadaļā "Categories", laukā "Add new category" norādot - "Name" - jaunās kategorijas nosaukumu | Paziņojums, ka kategorija ir pievienota pieejamo kategoriju sarakstam – ‘Category added.’ | Pareizi |
| 11. | Kategoriju noņemšana no pieejamo kategoriju saraksta, veidojot ziņu | Brīdinājums, vai mes tiešām vēlamies dzēst kategoriju. Jāspež "OK". Kategorija tiek izdzēsta | Pareizi |
| 12. | 7. | Jānodrošina lietotājam mainit savo paroli. | Ievadot jaunu paroli un pieļaujot kļūdu atkārtojot jauno paroli (paroles nesakrīt) | Paziņojums, ka parole un atkartota parole nesakrīt. “Password and confirm password not matched.” | Pareizi |
| 13. | Ievadot esošo paroli un pieļaujot kļūdu to ievadīšanai (parole neatbīlst esošam) | Paziņojums, ka esoša parole neatbīlst laukā norādītajai parolei. “Your current password is incorrect.” | Pareizi |

# NOBEIGUMS

Vietne pilda savus uzdevumus un mērķus, iespēja rakstīt pielāgotus rakstus un ziņas tiek īstenota pieņemamā formā, bet galvenais tas kas ir skaidrs kādā virzienā un kādos aspektos vietni var papildināt un uzlabot. Vietnes ideja ir interesanta, tāpēc pārliecināts ka to var tālāk attīstīt un izlaist vispārējai lietošanai internetā. Darba procesā ar vietnei tika secināts, ka mijedarbība un darba ar ietvaru ‘WordPress’ ir ļoti ērta un nav problemātiska, šis ir patiesi perspektīvs ietvars vietņu veidošanai. Bija iegūta vērtīga pieredze veidojot šo vietni. Tika sastādīts arī nerealizēto ideju saraksts attiecībā uz tehnisko pusi, piemēram:

* SQLi uz PDO ar bindParam, lai sagatavotu un izpildītu SQL priekšrakstu bez vietturiem un SQL injekcijas nebija iespējama. Datu bāzes un vietņu aizsardzība pret uzlaušanu.
* Koda formatēšana - PHP ir formatēšanas standarti, piemēram PSR-2, to var automātiski iepriekš instalēt koda redaktorā vai arī varat to iepriekš instalēt manuāli, lai automātiski formatētu kodu.

Nākotnē plānots paplašināt funkcionalitāti, datubāzē pievienot vairāk tabulu lielākai lietotāju mijiedarbībai ar vietni, pilnveidot dizainu un pilnveidot lietotāja profilu. Iespējams apgūt jaunu ietvaru un migrēt šo vietni uz pārāko.

# IZMANTOTIE AVOTI

1. Use-case pamācība priekš diagrammas izveidei - <https://www.lucidchart.com/pages/uml-use-case-diagram> - (Resurss apskatīts 27.01.2024.).
2. ER pamācība priekš diagrammas izveidei - <https://www.lucidchart.com/pages/er-diagrams> - (Resurss apskatīts 27.01.2024.).
3. Pamācība priekš projekta izveidei - [https://getbootstrap.com/docs/5.3](https://getbootstrap.com/docs/5.3/getting-started) - (Resurss apskatīts 20.02.2024.).
4. Pamacībā priekš darbošanai ar XAMPP - <https://habr.com/ru/articles/439994/> - (Resurss apskatīts 06.11.2023.).
5. Pamacībā priekš datubāzes izveidei ar phpMyAdmin - [phpMyAdmin - Quick Guide (tutorialspoint.com)](https://www.tutorialspoint.com/phpmyadmin/phpmyadmin_quick_guide.htm) - (Resurss apskatīts 10.09.2023.)
6. Pamacībā priekš funkcionālās dekompozīcijas diagrammas izveidei - <https://binaryterms.com/functional-decomposition.html> - (Resurss apskatīts 21.02.2024.)
7. Pamacībā priekš ‘WordPress’ vietnes izveidei - <https://wordpress.com/support/five-step-website-setup/> - (Resurss apskatīts 11.02.2024.)
8. Pamacībā priekš ‘WordPress’ vietnes izveidei -

<https://blog.hubspot.com/marketing/wordpress-website> - (Resurss apskatīts 11.02.2024.)

1. Pamacībā priekš ‘WordPress’ vietnes izveidei par ziņu un rakstu tēmai - <https://www.inmotionhosting.com/support/edu/wordpress/plugins/create-wordpress-news-website/> (Resurss apskatīts 13.02.2024.)
2. Paskaidrojums par to, kā vietņu migrēšanai izmantot WordPress spraudni ‘Dublicator’ (plugin) - <https://wordpress.org/plugins/duplicator/> (Resurss apskatīts 20.02.2024.)
3. Pamacībā priekš ‘WordPress’ spraudni ‘Frontend Post Submission Manager’ kas ļauj realizēt pašu svarīgāko funkcionalitāti – ziņu un rakstu veidošanu - <https://wpshuffle.com/wordpress-plugins/frontend-post-submission-manager/#:~:text=Frontend%20Post%20Submission%20Manager%20is,with%20Frontend%20Post%20Management%20Dashboard>. (Resurss apskatīts 20.02.2024.)
4. Pamacībā priekš ‘WordPress’ spraudni ‘Frontend Post Submission Manager’ kas ļauj realizēt funkcionalitāti veidot ziņas un rakstus - <https://www.youtube.com/watch?v=oOd3kbQjrKg&ab_channel=WPShuffle> (Resurss apskatīts 20.02.2024.)
5. Pamacībā priekš ‘WordPress’ spraudni ‘Ajax Load More’ kas ļauj pievienot pogu “load more” (ielādēt vairāk), lai saturs turpinātu parādīties plūsmā - <https://www.youtube.com/watch?v=W87m1XRgsGs&ab_channel=WTF> (Resurss apskatīts 20.02.2024.)
6. Pamacībā priekš ‘WordPress’ spraudni ‘User registration’ kas ļauj veidot registrācijas un autorizācijas formu kā arī veido vienkaršo profilu - <https://www.youtube.com/watch?v=jbiBJdowxG0&ab_channel=ThemeGrill> (Resurss apskatīts 21.02.2024.)
7. Pamacībā priekš ‘WordPress’ spraudni ‘advanced custom fields pro’ - <https://www.youtube.com/playlist?list=PLgFB6lmeXFOoyP8YUj7TmDnzFrrPcjCf3> (Resurss apskatīts 22.02.2024.)
8. Pamacībā priekš ‘WordPress’ spraudni ‘advanced custom fields pro’. Spraudni automatiski izveido tabulu wp\_term\_taxonomy datubāzē - <https://www.advancedcustomfields.com/resources/getting-started-with-acf/#:~:text=Advanced%20Custom%20Fields%20is%20a,and%20educate%20your%20clients%20quicker>. (Resurss apskatīts 22.02.2024.)

# PIELIKUMI

1. pielikums

**Datu bāzes izmešana**

-- phpMyAdmin SQL Dump

-- version 5.2.1

-- https://www.phpmyadmin.net/

--

-- Хост: 127.0.0.1

-- Время создания: Май 23 2024 г., 18:40

-- Версия сервера: 10.4.32-MariaDB

-- Версия PHP: 8.2.12

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- База данных: `ap-site`

--

-- --------------------------------------------------------

--

-- Структура таблицы `wp\_alm`

--

CREATE TABLE `wp\_alm` (

`id` mediumint(9) NOT NULL,

`name` text NOT NULL,

`repeaterDefault` longtext NOT NULL,

`repeaterType` text NOT NULL,

`pluginVersion` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп данных таблицы `wp\_alm`

--

INSERT INTO `wp\_alm` (`id`, `name`, `repeaterDefault`, `repeaterType`, `pluginVersion`) VALUES

(1, 'default', '  <div class=\"col s12 m12 l12 lenta-item\">\n <div class=\"card card-feed animated\">\n <div class=\"collection\">\n <div class=\"lenta\_cat blue\">\n <?php $cats = get\_the\_category\_list( \' \' );\n echo $cats; ?>\n </div> \n </div> \n <div class=\"card-content lenta\">\n <h4 itemprop=\"name\"><?php the\_title(); ?></h4>\n </div> \n <div class=\"card-image\">\n <a href=\"<?php the\_permalink(); ?>\"><img style=\"max-height: 300px;\" itemprop=\"image\" alt=\"<?php the\_title(); ?>\" src=\"<?php the\_post\_thumbnail\_url(\'full\'); ?>\"></a>\n </div>\n <div class=\"card-action flex no-border\">\n <a class=\"btn btn-flat z-depth-0 none-hover flex\"><i class=\"material-icons hide\" style=\"margin-right: 4px;\">remove\_red\_eye</i><?php echo get\_the\_date(\'d F\'); ?></a>\n <a class=\"btn btn-flat waves-effect z-depth-0 flex-right white-text\" href=\"<?php the\_permalink(); ?>\">Читать</a>\n \n \n </div>\n </div>\n </div>', 'default', '7.1.0.1');

-- --------------------------------------------------------

--

-- Структура таблицы `wp\_commentmeta`

--

CREATE TABLE `wp\_commentmeta` (

`meta\_id` bigint(20) UNSIGNED NOT NULL,

`comment\_id` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`meta\_key` varchar(255) DEFAULT NULL,

`meta\_value` longtext DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

-- --------------------------------------------------------

--

-- Структура таблицы `wp\_comments`

--

CREATE TABLE `wp\_comments` (

`comment\_ID` bigint(20) UNSIGNED NOT NULL,

`comment\_post\_ID` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`comment\_author` tinytext NOT NULL,

`comment\_author\_email` varchar(100) NOT NULL DEFAULT '',

`comment\_author\_url` varchar(200) NOT NULL DEFAULT '',

`comment\_author\_IP` varchar(100) NOT NULL DEFAULT '',

`comment\_date` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`comment\_date\_gmt` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`comment\_content` text NOT NULL,

`comment\_karma` int(11) NOT NULL DEFAULT 0,

`comment\_approved` varchar(20) NOT NULL DEFAULT '1',

`comment\_agent` varchar(255) NOT NULL DEFAULT '',

`comment\_type` varchar(20) NOT NULL DEFAULT 'comment',

`comment\_parent` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`user\_id` bigint(20) UNSIGNED NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Дамп данных таблицы `wp\_comments`

--

INSERT INTO `wp\_comments` (`comment\_ID`, `comment\_post\_ID`, `comment\_author`, `comment\_author\_email`, `comment\_author\_url`, `comment\_author\_IP`, `comment\_date`, `comment\_date\_gmt`, `comment\_content`, `comment\_karma`, `comment\_approved`, `comment\_agent`, `comment\_type`, `comment\_parent`, `user\_id`) VALUES

(1, 1, 'Комментатор WordPress', 'wapuu@wordpress.example', 'https://ru.wordpress.org/', '', '2024-04-02 21:51:09', '2024-04-02 18:51:09', 'Привет! Это комментарий.\nЧтобы начать модерировать, редактировать и удалять комментарии, перейдите на экран «Комментарии» в консоли.\nАватары авторов комментариев загружаются с сервиса <a href=\"https://ru.gravatar.com/\">Gravatar</a>.', 0, '1', '', 'comment', 0, 0);

-- --------------------------------------------------------

--

-- Структура таблицы `wp\_duplicator\_packages`

--

CREATE TABLE `wp\_duplicator\_packages` (

`id` bigint(20) UNSIGNED NOT NULL,

`name` varchar(250) NOT NULL,

`hash` varchar(50) NOT NULL,

`status` int(11) NOT NULL,

`created` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`owner` varchar(60) NOT NULL,

`package` longtext NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Структура таблицы `wp\_fpsm\_forms`

--

CREATE TABLE `wp\_fpsm\_forms` (

`form\_id` mediumint(9) NOT NULL,

`form\_title` varchar(255) DEFAULT NULL,

`form\_alias` varchar(255) DEFAULT NULL,

`form\_details` longtext DEFAULT NULL,

`post\_type` varchar(255) DEFAULT NULL,

`form\_type` varchar(255) DEFAULT NULL,

`form\_status` mediumint(9) NOT NULL DEFAULT 1

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Структура таблицы `wp\_links`

--

CREATE TABLE `wp\_links` (

`link\_id` bigint(20) UNSIGNED NOT NULL,

`link\_url` varchar(255) NOT NULL DEFAULT '',

`link\_name` varchar(255) NOT NULL DEFAULT '',

`link\_image` varchar(255) NOT NULL DEFAULT '',

`link\_target` varchar(25) NOT NULL DEFAULT '',

`link\_description` varchar(255) NOT NULL DEFAULT '',

`link\_visible` varchar(20) NOT NULL DEFAULT 'Y',

`link\_owner` bigint(20) UNSIGNED NOT NULL DEFAULT 1,

`link\_rating` int(11) NOT NULL DEFAULT 0,

`link\_updated` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`link\_rel` varchar(255) NOT NULL DEFAULT '',

`link\_notes` mediumtext NOT NULL,

`link\_rss` varchar(255) NOT NULL DEFAULT ''

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

-- --------------------------------------------------------

--

-- Структура таблицы `wp\_options`

--

CREATE TABLE `wp\_options` (

`option\_id` bigint(20) UNSIGNED NOT NULL,

`option\_name` varchar(191) NOT NULL DEFAULT '',

`option\_value` longtext NOT NULL,

`autoload` varchar(20) NOT NULL DEFAULT 'yes'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Дамп данных таблицы `wp\_options`

--

INSERT INTO `wp\_options` (`option\_id`, `option\_name`, `option\_value`, `autoload`) VALUES

(1, 'siteurl', 'http://localhost/ap-site', 'yes'),

(2, 'home', 'http://localhost/ap-site', 'yes'),

(3, 'blogname', 'Retell', 'yes'),

(4, 'blogdescription', '', 'yes'),

(5, 'users\_can\_register', '1', 'yes'),

(6, 'admin\_email', 'vt2360@gmail.com', 'yes'),

(7, 'start\_of\_week', '1', 'yes'),

(8, 'use\_balanceTags', '0', 'yes'),

(9, 'use\_smilies', '1', 'yes'),

(10, 'require\_name\_email', '1', 'yes'),

(11, 'comments\_notify', '1', 'yes'),

--

-- Структура таблицы `wp\_posts`

--

CREATE TABLE `wp\_posts` (

`ID` bigint(20) UNSIGNED NOT NULL,

`post\_author` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`post\_date` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`post\_date\_gmt` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`post\_content` longtext NOT NULL,

`post\_title` text NOT NULL,

`post\_excerpt` text NOT NULL,

`post\_status` varchar(20) NOT NULL DEFAULT 'publish',

`comment\_status` varchar(20) NOT NULL DEFAULT 'open',

`ping\_status` varchar(20) NOT NULL DEFAULT 'open',

`post\_password` varchar(255) NOT NULL DEFAULT '',

`post\_name` varchar(200) NOT NULL DEFAULT '',

`to\_ping` text NOT NULL,

`pinged` text NOT NULL,

`post\_modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`post\_modified\_gmt` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`post\_content\_filtered` longtext NOT NULL,

`post\_parent` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`guid` varchar(255) NOT NULL DEFAULT '',

`menu\_order` int(11) NOT NULL DEFAULT 0,

`post\_type` varchar(20) NOT NULL DEFAULT 'post',

`post\_mime\_type` varchar(100) NOT NULL DEFAULT '',

`comment\_count` bigint(20) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Дамп данных таблицы `wp\_posts`

--

INSERT INTO `wp\_posts` (`ID`, `post\_author`, `post\_date`, `post\_date\_gmt`, `post\_content`, `post\_title`, `post\_excerpt`, `post\_status`, `comment\_status`, `ping\_status`, `post\_password`, `post\_name`, `to\_ping`, `pinged`, `post\_modified`, `post\_modified\_gmt`, `post\_content\_filtered`, `post\_parent`, `guid`, `menu\_order`, `post\_type`, `post\_mime\_type`, `comment\_count`) VALUES

(1, 1, '2024-04-02 21:51:09', '2024-04-02 18:51:09', '<!-- wp:paragraph -->\r\n<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p>\r\n<!-- /wp:paragraph -->', 'Ирландская мафия', '', 'publish', 'open', 'open', '', 'privet-mir', '', '', '2024-04-03 19:57:42', '2024-04-03 16:57:42', '', 0, 'http://localhost/ap-site/?p=1', 0, 'post', '', 1),

--

-- Структура таблицы `wp\_termmeta`

--

CREATE TABLE `wp\_termmeta` (

`meta\_id` bigint(20) UNSIGNED NOT NULL,

`term\_id` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`meta\_key` varchar(255) DEFAULT NULL,

`meta\_value` longtext DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

-- --------------------------------------------------------

--

-- Структура таблицы `wp\_terms`

--

CREATE TABLE `wp\_terms` (

`term\_id` bigint(20) UNSIGNED NOT NULL,

`name` varchar(200) NOT NULL DEFAULT '',

`slug` varchar(200) NOT NULL DEFAULT '',

`term\_group` bigint(10) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Дамп данных таблицы `wp\_terms`

--

INSERT INTO `wp\_terms` (`term\_id`, `name`, `slug`, `term\_group`) VALUES

(1, 'Without category', 'bez-rubriki', 0),

(2, 'Menu 1', 'menu-1', 0),

(3, 'Different', 'raznoe', 0),

(4, 'Astronomy', 'astronomiya', 0),

(5, 'History', 'istoriya', 0),

(6, 'Popular', 'populyarnoe', 0),

(7, 'NEWS', 'novosti', 0);

-- --------------------------------------------------------

--

-- Структура таблицы `wp\_term\_relationships`

--

CREATE TABLE `wp\_term\_relationships` (

`object\_id` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`term\_taxonomy\_id` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`term\_order` int(11) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Дамп данных таблицы `wp\_term\_relationships`

--

INSERT INTO `wp\_term\_relationships` (`object\_id`, `term\_taxonomy\_id`, `term\_order`) VALUES

(1, 5, 0),

(1, 6, 0),

(16, 3, 0),

--

-- Структура таблицы `wp\_term\_taxonomy`

--

CREATE TABLE `wp\_term\_taxonomy` (

`term\_taxonomy\_id` bigint(20) UNSIGNED NOT NULL,

`term\_id` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`taxonomy` varchar(32) NOT NULL DEFAULT '',

`description` longtext NOT NULL,

`parent` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`count` bigint(20) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Дамп данных таблицы `wp\_term\_taxonomy`

--

INSERT INTO `wp\_term\_taxonomy` (`term\_taxonomy\_id`, `term\_id`, `taxonomy`, `description`, `parent`, `count`) VALUES

(1, 1, 'category', '', 0, 0),

--

-- Структура таблицы `wp\_usermeta`

--

CREATE TABLE `wp\_usermeta` (

`umeta\_id` bigint(20) UNSIGNED NOT NULL,

`user\_id` bigint(20) UNSIGNED NOT NULL DEFAULT 0,

`meta\_key` varchar(255) DEFAULT NULL,

`meta\_value` longtext DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Дамп данных таблицы `wp\_usermeta`

--

INSERT INTO `wp\_usermeta` (`umeta\_id`, `user\_id`, `meta\_key`, `meta\_value`) VALUES

(1, 1, 'nickname', 'admin'),

(2, 1, 'first\_name', ''),

(3, 1, 'last\_name', ''),

(4, 1, 'description', ''),

(5, 1, 'rich\_editing', 'true'),

--

-- Структура таблицы `wp\_users`

--

CREATE TABLE `wp\_users` (

`ID` bigint(20) UNSIGNED NOT NULL,

`user\_login` varchar(60) NOT NULL DEFAULT '',

`user\_pass` varchar(255) NOT NULL DEFAULT '',

`user\_nicename` varchar(50) NOT NULL DEFAULT '',

`user\_email` varchar(100) NOT NULL DEFAULT '',

`user\_url` varchar(100) NOT NULL DEFAULT '',

`user\_registered` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`user\_activation\_key` varchar(255) NOT NULL DEFAULT '',

`user\_status` int(11) NOT NULL DEFAULT 0,

`display\_name` varchar(250) NOT NULL DEFAULT ''

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Дамп данных таблицы `wp\_users`

--

INSERT INTO `wp\_users` (`ID`, `user\_login`, `user\_pass`, `user\_nicename`, `user\_email`, `user\_url`, `user\_registered`, `user\_activation\_key`, `user\_status`, `display\_name`) VALUES

(1, 'admin', '$P$BMpGLwtuOl5vD7bVTsdUx3qXHKKYil1', 'admin', 'vt2360@gmail.com', 'http://localhost/ap-site', '2024-04-02 18:51:07', '', 0, 'admin'),

--

-- Структура таблицы `wp\_user\_registration\_sessions`

--

CREATE TABLE `wp\_user\_registration\_sessions` (

`session\_id` bigint(20) UNSIGNED NOT NULL,

`session\_key` char(32) NOT NULL,

`session\_value` longtext NOT NULL,

`session\_expiry` bigint(20) UNSIGNED NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_520\_ci;

--

-- Индексы сохранённых таблиц

--

--

-- Индексы таблицы `wp\_alm`

--

ALTER TABLE `wp\_alm`

ADD UNIQUE KEY `id` (`id`);

--

-- Индексы таблицы `wp\_commentmeta`

--

ALTER TABLE `wp\_commentmeta`

ADD PRIMARY KEY (`meta\_id`),

ADD KEY `comment\_id` (`comment\_id`),

ADD KEY `meta\_key` (`meta\_key`(191));

--

-- Индексы таблицы `wp\_comments`

--

ALTER TABLE `wp\_comments`

ADD PRIMARY KEY (`comment\_ID`),

ADD KEY `comment\_post\_ID` (`comment\_post\_ID`),

ADD KEY `comment\_approved\_date\_gmt` (`comment\_approved`,`comment\_date\_gmt`),

ADD KEY `comment\_date\_gmt` (`comment\_date\_gmt`),

ADD KEY `comment\_parent` (`comment\_parent`),

ADD KEY `comment\_author\_email` (`comment\_author\_email`(10));

--

-- Индексы таблицы `wp\_duplicator\_packages`

--

ALTER TABLE `wp\_duplicator\_packages`

ADD PRIMARY KEY (`id`),

ADD KEY `hash` (`hash`);

--

-- Индексы таблицы `wp\_fpsm\_forms`

--

ALTER TABLE `wp\_fpsm\_forms`

ADD PRIMARY KEY (`form\_id`);

--

-- Индексы таблицы `wp\_links`

--

ALTER TABLE `wp\_links`

ADD PRIMARY KEY (`link\_id`),

ADD KEY `link\_visible` (`link\_visible`);

--

-- Индексы таблицы `wp\_options`

--

ALTER TABLE `wp\_options`

ADD PRIMARY KEY (`option\_id`),

ADD UNIQUE KEY `option\_name` (`option\_name`),

ADD KEY `autoload` (`autoload`);

--

-- Индексы таблицы `wp\_postmeta`

--

ALTER TABLE `wp\_postmeta`

ADD PRIMARY KEY (`meta\_id`),

ADD KEY `post\_id` (`post\_id`),

ADD KEY `meta\_key` (`meta\_key`(191));

--

-- Индексы таблицы `wp\_posts`

--

ALTER TABLE `wp\_posts`

ADD PRIMARY KEY (`ID`),

ADD KEY `post\_name` (`post\_name`(191)),

ADD KEY `type\_status\_date` (`post\_type`,`post\_status`,`post\_date`,`ID`),

ADD KEY `post\_parent` (`post\_parent`),

ADD KEY `post\_author` (`post\_author`);

--

-- Индексы таблицы `wp\_termmeta`

--

ALTER TABLE `wp\_termmeta`

ADD PRIMARY KEY (`meta\_id`),

ADD KEY `term\_id` (`term\_id`),

ADD KEY `meta\_key` (`meta\_key`(191));

--

-- Индексы таблицы `wp\_terms`

--

ALTER TABLE `wp\_terms`

ADD PRIMARY KEY (`term\_id`),

ADD KEY `slug` (`slug`(191)),

ADD KEY `name` (`name`(191));

--

-- Индексы таблицы `wp\_term\_relationships`

--

ALTER TABLE `wp\_term\_relationships`

ADD PRIMARY KEY (`object\_id`,`term\_taxonomy\_id`),

ADD KEY `term\_taxonomy\_id` (`term\_taxonomy\_id`);

--

-- Индексы таблицы `wp\_term\_taxonomy`

--

ALTER TABLE `wp\_term\_taxonomy`

ADD PRIMARY KEY (`term\_taxonomy\_id`),

ADD UNIQUE KEY `term\_id\_taxonomy` (`term\_id`,`taxonomy`),

ADD KEY `taxonomy` (`taxonomy`);

--

-- Индексы таблицы `wp\_usermeta`

--

ALTER TABLE `wp\_usermeta`

ADD PRIMARY KEY (`umeta\_id`),

ADD KEY `user\_id` (`user\_id`),

ADD KEY `meta\_key` (`meta\_key`(191));

--

-- Индексы таблицы `wp\_users`

--

ALTER TABLE `wp\_users`

ADD PRIMARY KEY (`ID`),

ADD KEY `user\_login\_key` (`user\_login`),

ADD KEY `user\_nicename` (`user\_nicename`),

ADD KEY `user\_email` (`user\_email`);

--

-- Индексы таблицы `wp\_user\_registration\_sessions`

--

ALTER TABLE `wp\_user\_registration\_sessions`

ADD PRIMARY KEY (`session\_key`),

ADD UNIQUE KEY `session\_id` (`session\_id`);

--

-- AUTO\_INCREMENT для сохранённых таблиц

--

--

-- AUTO\_INCREMENT для таблицы `wp\_alm`

--

ALTER TABLE `wp\_alm`

MODIFY `id` mediumint(9) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT для таблицы `wp\_commentmeta`

--

ALTER TABLE `wp\_commentmeta`

MODIFY `meta\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT для таблицы `wp\_comments`

--

ALTER TABLE `wp\_comments`

MODIFY `comment\_ID` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT для таблицы `wp\_duplicator\_packages`

--

ALTER TABLE `wp\_duplicator\_packages`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT для таблицы `wp\_fpsm\_forms`

--

ALTER TABLE `wp\_fpsm\_forms`

MODIFY `form\_id` mediumint(9) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT для таблицы `wp\_links`

--

ALTER TABLE `wp\_links`

MODIFY `link\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT для таблицы `wp\_options`

--

ALTER TABLE `wp\_options`

MODIFY `option\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=620;

--

-- AUTO\_INCREMENT для таблицы `wp\_postmeta`

--

ALTER TABLE `wp\_postmeta`

MODIFY `meta\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=252;

--

-- AUTO\_INCREMENT для таблицы `wp\_posts`

--

ALTER TABLE `wp\_posts`

MODIFY `ID` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=97;

--

-- AUTO\_INCREMENT для таблицы `wp\_termmeta`

--

ALTER TABLE `wp\_termmeta`

MODIFY `meta\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT для таблицы `wp\_terms`

--

ALTER TABLE `wp\_terms`

MODIFY `term\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

--

-- AUTO\_INCREMENT для таблицы `wp\_term\_taxonomy`

--

ALTER TABLE `wp\_term\_taxonomy`

MODIFY `term\_taxonomy\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

--

-- AUTO\_INCREMENT для таблицы `wp\_usermeta`

--

ALTER TABLE `wp\_usermeta`

MODIFY `umeta\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=51;

--

-- AUTO\_INCREMENT для таблицы `wp\_users`

--

ALTER TABLE `wp\_users`

MODIFY `ID` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT для таблицы `wp\_user\_registration\_sessions`

--

ALTER TABLE `wp\_user\_registration\_sessions`

MODIFY `session\_id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

**Vietnes kods**

1. pielikums

**ap-site/wp-content/themes/retell/404.php**

<?php get\_template\_part('includes/header'); ?>

<main class="animated">

<div class="container">

<div class="section">

<div class="row">

<div class="col s12 center animated"></div>

<div class="col s12 m6 l4 center animated "><img style="width: 100%; height: 100%;" src="<?php echo get\_template\_directory\_uri() ?>/img/404.png"></div>

<div class="col s12 m6 l4">

<h4 class="grey-text text-darken-2 center animated ">

<div class="hide-on-small-and-down">

<br><div class="hide-on-med-and-down"><br></div>

СPage not found

<br><br>

</div>

<a href="<?php echo site\_url(); ?>" class="btn btn-large blue accent-4 waves-effect animated opacity-zero">To the home page</a>

</h4>

</div>

</div>

</div>

</div>

</main>

<?php get\_template\_part('includes/footer'); ?>

**ap-site/wp-content/themes/retell/category.php**

<?php get\_template\_part('includes/header'); ?>

<main class="animated">

<div style="padding-bottom: 40px; padding-top: 20px">

<div class="container">

<div class="section">

<div class="row">

<div class="col s12 animated">

<h5 class="light grey-text">#<?php echo single\_cat\_title(); ?></h5>

<br>

</div>

<div class="col s12 l8">

<div class="row" id="article-list">

<?php if(have\_posts()): while(have\_posts()): the\_post(); ?>

<div class="col s12 m12 l12 lenta-item">

<div class="card card-feed animated">

<div class="collection">

<div class="lenta\_cat blue">

<?php $cats = get\_the\_category\_list( ' ' );

echo $cats; ?>

</div>

</div>

<div class="card-content lenta">

<h4 itemprop="name"><?php the\_title(); ?></h4>

</div>

<div class="card-image">

<a href="<?php the\_permalink(); ?>"><img style="max-height: 300px;" itemprop="image" alt="<?php the\_title(); ?>" src="<?php the\_post\_thumbnail\_url('full'); ?>"></a>

</div>

<div class="card-action flex no-border">

<a class="btn btn-flat z-depth-0 none-hover flex"><i class="material-icons hide" style="margin-right: 4px;">remove\_red\_eye</i><?php echo get\_the\_date('d F'); ?></a>

<a class="btn btn-flat waves-effect z-depth-0 flex-right white-text" href="<?php the\_permalink(); ?>">Read</a>

</div>

</div>

</div>

<?php endwhile; ?>

<?php endif; ?>

<?php if ( function\_exists('bst\_pagination') ) { bst\_pagination(); } ?>

</div>

</div>

<div class="col s12 l4">

<div class="row">

<div class="col s12">

<div class="collection card transparent animated cats" style="margin-top: 22px;">

<h4 style="margin-left: 56px;">Categories</h4>

<ul>

<?php wp\_list\_categories('orderby=name&exclude=1&title\_li='); ?>

</ul>

</div>

</div>

<div class="col s12">

</div>

</div>

</div>

</div>

</div>

</div></main>

<?php get\_template\_part('includes/footer'); ?>

**ap-site/wp-content/themes/retell/front-page.php**

<?php get\_template\_part('includes/header'); ?>

<main class="animated">

<div style="padding-bottom: 40px; padding-top: 20px" class="index-bg-color">

<div class="container">

<div class="section">

<div class="row">

<div class="col s12 l7">

<div class="row">

<div class="col s12 animated"><h5 class="light grey-text"><?php the\_field('main\_block','options'); ?></h5><br></div>

<?php $args = array('post\_type' => 'post',

'posts\_per\_page' =>6,

'post\_status' => 'publish',

'order' => 'DESC',

);

$wp\_query = new WP\_Query( $args );

if ( $wp\_query->have\_posts() ) {

while ( $wp\_query->have\_posts() ) {

$wp\_query->the\_post(); ?>

<div class="col s12 m6 l6">

<div class="card animated">

<a href="<?php the\_permalink(); ?>" style="display: block" class="card-image">

<img loading="lazy" itemprop="image" alt="<?php the\_title(); ?>" style="height: 200px; object-fit: cover;" src="<?php the\_post\_thumbnail\_url('full'); ?>">

<span class="card-title" itemprop="name"><?php the\_title(); ?></span>

</a>

<div class="card-action flex no-border">

<a class="btn btn-flat z-depth-0 none-hover"><?php echo get\_the\_date('d F'); ?></a>

<a href="<?php the\_permalink(); ?>" class="btn btn-flat waves-effect z-depth-0 flex-right white-text">Read</a>

</div>

</div>

</div>

<?php } }

wp\_reset\_query(); ?>

</div>

</div>

<div class="col s12 l1"></div>

<div class="col s12 l4">

<div class="row" style="padding-bottom: 48px">

<div class="col s12 animated">

<h5 class="light grey-text"><?php the\_field('right\_block','options'); ?></h5>

<br>

</div>

<div class="col s12">

<?php $args = array('post\_type' => 'post',

'posts\_per\_page' =>5,

'post\_status' => 'publish',

'orderby' => 'rand',

);

$wp\_query = new WP\_Query( $args );

if ( $wp\_query->have\_posts() ) {

while ( $wp\_query->have\_posts() ) {

$wp\_query->the\_post(); ?>

<div class="card horizontal animated">

<div class="card-image">

<a href="<?php the\_permalink(); ?>"><img itemprop="image" alt="<?php the\_title(); ?>" src="<?php the\_post\_thumbnail\_url('full'); ?>"></a>

</div>

<div class="card-stacked">

<a href="<?php the\_permalink(); ?> " class="card-content bw-text">

<p itemprop="name"><?php the\_title(); ?></p>

</a>

</div>

</div>

<?php } }

wp\_reset\_query(); ?>

</div>

</div>

</div>

<div class="col s12">

<div class="row">

<div class="col s12 animated"><h5 class="light grey-text"><a class="grey-text" href="articles"><?php the\_field('bottom\_block','options'); ?></a></h5><br></div>

<?php $args = array('post\_type' => 'post',

'cat' => 6,

'posts\_per\_page' =>12,

'post\_status' => 'publish',

'order' => 'DESC',

);

$wp\_query = new WP\_Query( $args );

if ( $wp\_query->have\_posts() ) {

while ( $wp\_query->have\_posts() ) {

$wp\_query->the\_post(); ?>

<div class="col s12 m6 l4">

<div class="card animated">

<a href="<?php the\_permalink(); ?>" style="display: block" class="card-image">

<img loading="lazy" itemprop="image" alt="<?php the\_title(); ?>" style="height: 200px; object-fit: cover;" src="<?php the\_post\_thumbnail\_url('full'); ?>">

<span class="card-title" itemprop="name"><?php the\_title(); ?></span>

</a>

<div class="card-action flex no-border">

<a class="btn btn-flat z-depth-0 none-hover"><?php echo get\_the\_date('d F'); ?></a>

<a href="<?php the\_permalink(); ?>" class="btn btn-flat waves-effect z-depth-0 flex-right white-text">Read</a>

</div>

</div>

</div>

<?php } }

wp\_reset\_query(); ?>

</div>

</div>

</div>

</div>

</div>

</div></main>

<?php get\_template\_part('includes/footer'); ?>

**ap-site/wp-content/themes/retell/functions.php**

<?php

/\*

All the functions are in the PHP pages in the functions/ folder.

\*/

require\_once locate\_template('/functions/cleanup.php');

require\_once locate\_template('/functions/setup.php');

require\_once locate\_template('/functions/enqueues.php');

require\_once locate\_template('/functions/navbar.php');

require\_once locate\_template('/functions/widgets.php');

require\_once locate\_template('/functions/search.php');

require\_once locate\_template('/functions/feedback.php');

add\_action('after\_setup\_theme', 'true\_load\_theme\_textdomain');

function true\_load\_theme\_textdomain(){

load\_theme\_textdomain( 'bst', get\_template\_directory() . '/languages' );

}

// Отключает новый редактор блоков в WordPress (Гутенберг).

if( 'disable\_gutenberg' ){

add\_filter( 'use\_block\_editor\_for\_post\_type', '\_\_return\_false', 100 );

remove\_action( 'wp\_enqueue\_scripts', 'wp\_common\_block\_scripts\_and\_styles' );

add\_action( 'admin\_init', function(){

remove\_action( 'admin\_notices', [ 'WP\_Privacy\_Policy\_Content', 'notice' ] );

add\_action( 'edit\_form\_after\_title', [ 'WP\_Privacy\_Policy\_Content', 'notice' ] );

} );

}

# Отключает редактор блоков для виджетов WordPress

add\_filter( 'use\_widgets\_block\_editor', '\_\_return\_false' );

add\_action('acf/init', 'my\_acf\_op\_init');

function my\_acf\_op\_init() {

// Check function exists.

if( function\_exists('acf\_add\_options\_page') ) {

// Register options page.

$option\_page = acf\_add\_options\_page(array(

'page\_title' => \_\_('Опции'),

'menu\_title' => \_\_('Опции'),

'menu\_slug' => 'theme-general-settings',

'capability' => 'edit\_posts',

'redirect' => false

));

}

}

**ap-site/wp-content/themes/retell/index.php**

<?php get\_template\_part('includes/header'); ?>

<div class="container">

<div id="content" role="main">

<?php get\_template\_part('includes/loops/content', get\_post\_format()); ?>

</div><!-- /#content -->

</div><!-- /.container -->

<?php get\_template\_part('includes/footer'); ?>

**ap-site/wp-content/themes/retell/page.php**

<?php get\_template\_part('includes/header'); ?>

<div class="container">

<div id="content" role="main">

<?php get\_template\_part('includes/loops/content', 'page'); ?>

</div><!-- /#content -->

</div><!-- /.container -->

<?php get\_template\_part('includes/footer'); ?>

***ap-site/wp-content/themes/retell/search.php***

<?php get\_template\_part('includes/header'); ?>

<main class="animated">

<div style="padding-bottom: 40px; padding-top: 20px">

<div class="container">

<div class="section">

<div class="row">

<div class="col s12 animated opacity-zero">

<h4 class="light grey-text flex">Search results: &ldquo;<?php the\_search\_query(); ?>&rdquo;</h4>

<br>

</div>

<?php if(have\_posts()): while(have\_posts()): the\_post(); ?>

<div class="col s12 m6 l4">

<div class="card animated opacity-zero">

<a href="<?php the\_permalink(); ?>" style="display: block" class="card-image">

<img loading="lazy" itemprop="image" alt="<?php the\_title(); ?>" style="height: 200px; object-fit: cover;" src="<?php the\_post\_thumbnail\_url('full'); ?>">

<span class="card-title" itemprop="name"><?php the\_title(); ?></span>

<div class="white-text card-tag blue-grey"><?php $cats = get\_the\_category\_list( ', ' );

echo $cats; ?></div>

<div class="card-action flex no-border">

<a class="btn btn-flat z-depth-0 none-hover"><?php echo get\_the\_date('d F'); ?></a>

<a href="<?php the\_permalink(); ?>" class="btn btn-flat waves-effect z-depth-0 flex-right white-text">Read</a>

</div>

</div>

</div>

<?php endwhile; else: ?>

<div class="alert alert-warning animated" style="margin-left: 14px; font-size: 18px;">

<i class="glyphicon glyphicon-exclamation-sign"></i> <?php \_e('Sorry, your search yielded no results.', 'bst'); ?>

</div>

<?php endif; ?>

</div>

</div>

</div>

</div></main>

<?php get\_template\_part('includes/footer'); ?>

**ap-site/wp-content/themes/retell/single.php**

<?php get\_template\_part('includes/header'); ?>

<main class="animated">

<div tgprop="article"><img itemprop="image" alt="<?php the\_title(); ?>" class="article-full-img animated opacity-zero" src="<?php the\_post\_thumbnail\_url('full'); ?>">

<div class="container article-color" style="max-width: 850px">

<div class="section">

<div class="row">

<div class="col s12">

<h4 itemprop="name" class="animated"><?php the\_title(); ?></h4>

<br>

<p><?php the\_content(); ?></p>

<br>

<div class="grey-text" style="font-size: 1.2rem;"><div class="row">

<div class="animated col s6 m4"><?php echo get\_the\_date('d F'); ?></div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</main>

<?php get\_template\_part('includes/footer'); ?>

**ap-site/wp-content/themes/retell/template\_lenta.php**

<?php

/\*\*

\* Template Name: Lenta

\*

\*/

get\_template\_part('includes/header'); ?>

<main class="animated">

<div style="padding-bottom: 40px; padding-top: 20px">

<div class="container">

<div class="section">

<div class="row">

<div class="col s12 animated">

<h5 class="light grey-text">#<?php the\_title(); ?></h5>

<br>

</div>

<div class="col s12 l8">

<div class="row" id="article-list">

<?php

$current = (get\_query\_var('paged')) ? get\_query\_var('paged') : 1;

$args = array('post\_type' => 'post',

'posts\_per\_page' =>4,

'post\_status' => 'publish',

'order' => 'DESC',

'paged' => $current

);

$wp\_query = new WP\_Query( $args );

if(have\_posts()): while(have\_posts()): the\_post(); ?>

<div class="col s12 m12 l12 lenta-item">

<div class="card card-feed animated">

<div class="collection">

<div class="lenta\_cat blue">

<?php $cats = get\_the\_category\_list( ' ' );

echo $cats; ?>

</div>

</div>

<div class="card-content lenta">

<h4 itemprop="name"><?php the\_title(); ?></h4>

</div>

<div class="card-image">

<a href="<?php the\_permalink(); ?>"><img style="max-height: 300px;" itemprop="image" alt="<?php the\_title(); ?>" src="<?php the\_post\_thumbnail\_url('full'); ?>"></a>

</div>

<div class="card-action flex no-border">

<a class="btn btn-flat z-depth-0 none-hover flex"><i class="material-icons hide" style="margin-right: 4px;">remove\_red\_eye</i><?php echo get\_the\_date('d F'); ?></a>

<a class="btn btn-flat waves-effect z-depth-0 flex-right white-text" href="<?php the\_permalink(); ?>">Read</a>

</div>

</div>

</div>

<?php endwhile; ?>

<?php endif; ?>

<?php echo do\_shortcode('[ajax\_load\_more post\_type="post" posts\_per\_page="2" offset="4" button\_label="Load more" button\_loading\_label="Loading..." button\_done\_label="No more posts" transition\_delay="100" pause="true"]'); ?>

<?php // if ( function\_exists('bst\_pagination') ) { bst\_pagination(); } ?>

</div>

</div>

<div class="col s12 l4">

<div class="row">

<div class="col s12">

<div class="collection card transparent animated cats" style="margin-top: 22px;">

<h4 style="margin-left: 56px;">Categories</h4>

<ul>

<?php wp\_list\_categories('orderby=name&exclude=1&title\_li='); ?>

</ul>

</div>

</div>

<div class="col s12">

</div>

</div>

</div>

</div>

</div>

</div></main>

<?php get\_template\_part('includes/footer'); ?>