

RĪGAS VALSTS TEHNIKUMS

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

KVALIFIKĀCIJAS DARBS

“Programmetāju tiešsaistes komunikēšanas platforma”

Paskaidrojošais raksts 67 lpp.

Audzēknis:

Mets Latuskins

Prakses vadītājs:

Igors Litvjakovs

Nodaļas vadītājs:

Normunds Barbāns

Rīga 2024

ANOTĀCIJA

Šajā kvalifikācijas darbā tiek prezentēts un analizēts tīmeklī balstītas sistēmas izstrādes process, kura mērķis ir pilnveidot izglītības un komunikācijas procesus, īpašu uzmanību pievēršot “komunicēšanas platformai”, kas ir sistēmas pasūtītājs. sistēma ir izstrādāta, izmantojot php, javascript un html/css valodas un sql datubāzi.

Darba struktūra ietver ievaddaļu, problēmas izklāstu, programmatūras risinājuma izvēles pamatojumu, modelēšanu un projektēšanu, datu struktūras aprakstu, lietotāja rokasgrāmatu, secinājumu, informācijas avotus un lietojumus.

Ievadā aprakstīta problēma un sistēmas atbilstība, un problēmas izklāsts sniedz īsu sistēmas galveno mērķu un funkcionalitātes aprakstu. Prasību specifikācija definē ievades un izvades informācijas aprakstus un iedala prasības funkcionālās un nefunkcionālās kategorijās.

Datu struktūras aprakstā ir detalizēti aprakstīti datu bāzes tabulu lauki un to attiecības. Lietotāja rokasgrāmatā ir ietverta informācija par sistēmas prasībām, instalēšana, programmas apraksts un testēšanas piemēri.

Kopumā šis darbs sniedz priekšstatu par tīmekļa sistēmas izstrādes procesu, koncentrējoties uz “Online Platformas” specifiskajām vajadzībām un prasībām, un ietver plašu informāciju par sistēmas izveidi un darbību, kā arī sniedz skaidra izpratne par izvēlētajām tehnoloģijām un to pielietojumu.

Kvalifikācijas darbs sastāv 67 lappusēm, 28 attēliem, 13 tabulām un 8 pielikumiem.

ANNOTATION

This qualification thesis presents and analyzes the process of developing a web-based system, which aims to improve education and communication processes, paying special attention to the "communication platform", which is the customer of the system. the system is developed using php, javascript and html/css languages and mysql database.

The structure of the paper includes introduction, problem statement, rationale for software solution selection, modeling and design, data structure description, user manual, conclusion, information sources and applications.

The introduction describes the problem and the suitability of the system, and the problem statement provides a brief description of the main goals and functionality of the system. A requirements specification defines descriptions of input and output information and divides requirements into functional and non-functional categories.

The data structure description describes in detail the fields of the database tables and their relationships. The user guide contains information about system requirements, installation, program description and testing examples.

In general, this work provides an overview of the web system development process, focusing on the specific needs and requirements of the "Online Platform", and includes extensive information on the system's creation and operation, as well as a clear understanding of the selected technologies and their application. The qualification work consists of 67 pages, 28 images, 13 tables and 8 appendices.

SATURS

IEVADS

1. UZDEVUMA NOSTĀDNE

2. PRASĪBU SPECIFIKĀCIJA

2.1 Ieejas un izejas informācijas apraksts

2.2 Funkcionālās prasības

2.3 nefunkcionālās prasības

3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS____

4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

4.1 Sistēmas struktūras modelis

4.2 Sistēmas ER modelis

4.3 Funkcionālais sistēmas modelis

5. DATU STRŪKTURU APRAKSTS

6. LIETOTĀJA CEĻVEDIS

6.1 Sistēmas prasības aparatūrai un programmatūrai

6.2 Sistēmas instalācija un palaišana

6. LIETOTĀJA CEĻVEDIS

7. PPROGRAMMAS APRAKSTS

NOBEIGUMS

INFORMACIJAS AVOTI

PIELIKUMI

IEVADS

Šobrīd ārkārtīgi svarīgi ir izmantot modernās tehnoloģijas, lai uzlabotu izglītības procesus un uzlabotu darba efektivitāti. Arvien pieaugoša tehnoloģiskā attīstība, kas notiek katru dienu, sniedz iespējas uzlabot mācību pieredzi, padarot to pieejamāku visiem dalībniekiem – gan studentiem, gan pieredzējušiem programmētājiem.

Tāpēc mūsdienu IT sistēmās esošo problēmu risināšanai tiek veidota "Programmētāju komunikācijas platforma".

Salīdzinot ar esošajiem konkurentiem, tādiem kā "stack-overflow" un "CodeForum", mūsu izstrādātā sistēma piedāvās būtiskus uzlabojumus un jaunas funkcijas, kas atbilst mūsdienu IT sfēras prasībām, tai skaitā iespēju viegli atrast sev vēlamo tēmu un tādējādi atrisināt problēmas. problēma ātrāk.

Biznesa analīze skaidri parāda, ka lietotāji (gan studenti, gan parastie programmētāji) gūs labumu no sistēmas, izmantojot daudzpusīgu un lietotājam draudzīgu rīku komplektu, kas uzlabos kodēšanas procesu, veicinot efektīvāku sadarbību un komunikāciju. Tādējādi piedāvāju ne tikai risinājumu, bet arī ieguvumu sabiedrībai kopumā - uzlabojumu strauji augošajā un ārkārtīgi pieprasītajā IT nozarē.

1. UZDEVUMA NOSTĀDNE

Kvalifikācijas darba uzdevums ir radīt "Tiešsaistes un komunikācijas platformu", kas apvieno mūsdienīgas tehnoloģijas ar efektīvu problēmu risināšanu un komunikācijas procesu organizēšanu.

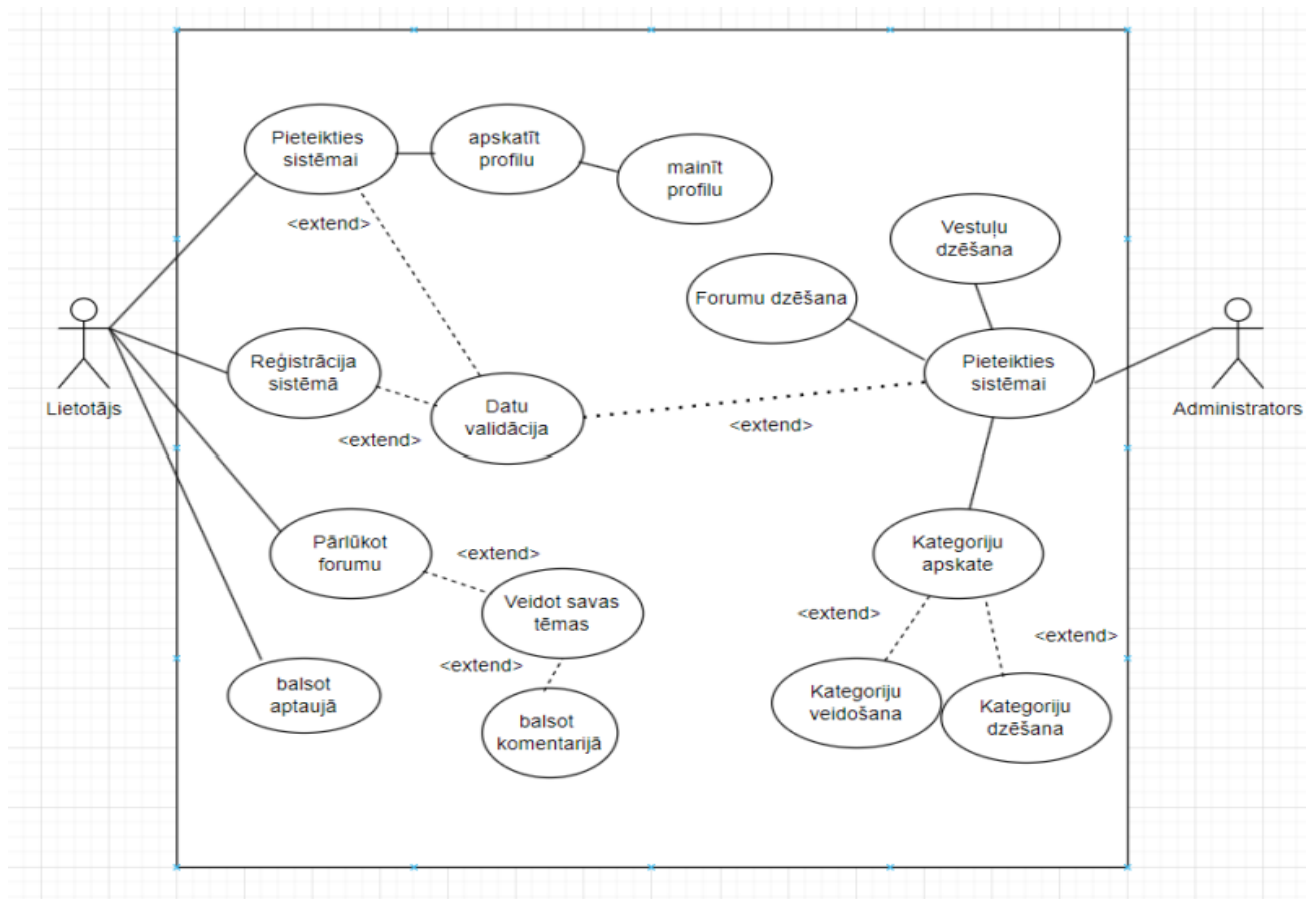
Platforma būs nepieciešama, lai nodrošinātu nepārtrauktu komunikācijas procesu, kas būs aktuāls tik ilgi, kamēr pastāvēs mācību vajadzība. Tās galvenā mērķauditorija būs studenti

un IT sfēras darbnieki.

Izglītības platforma iekļaus šādas jauninājumus un funkcijas:

- Reģistrācija un pierakstīšana lai nodrošinātu personalizētu piekļuvi uz vienti.
- Dažādas piekļuves tiesības, ko nodrošina administratori, atkarībā no lietotāja lomas un nepieciešamības.
- Failu attēlošanas sistēma serverī, kas ļauj apskatīt ielādētus attēlus.
- Forumā ar iespējām uzdot jautājumus, komentēt, kā arī administrācijai dzēst vai rediģēt lietotāju ziņojumus.
- Satura kategorizēšana, lai lietotājiem būtu ērtāk atrast nepieciešamo informāciju.

Paredzēts, ka sistēmu lietos trīs lietotāju klases: Lietotājs(Lietotājs), un Administrators(skat.1.1.att).



1.1.att. Lietojumgadījuma diagramma

2. PRASĪBU SPECIFIKĀCIJA

2.1 Ieejas un izejas informācijas apraksts

Sistēmā tiks nodrošināta šādas ieejas informācijas apstrāde.

1. **Lietotāja dati** tiks ievadīti sistēmā, kad lietotājs reģistrēsies vai pieteiksies sistēmai.

Šie dati ietver:

2. **Vārds** – lietotāja vārds – burtu teksts ar izmēru līdz 20 rakstzīmēm. (Piem. “Mets”)
3. **E-pasts** – lietotāja e-pasts – burtu teksts ar izmēru līdz 20 rakstzīmēm. (Piem. “Mets@gmail.com”)
4. **Lietotājvārds** – lietotāja lietotājvārds – burtu teksts ar izmēru līdz 40 rakstzīmēm. (Piem. “MetsMets”)
5. **Parole** – lietotāja parole – burtu teksts ar izmēru līdz 255 rakstzīmēm. (Piem. “ManaParole123”)
6. Lietotājvārds, vārds, e-pasts un parole tiks ievadīti no klaviatūras.
7. **Jautājuma dati** tiks ievadīti sistēmā, kad lietotājs izveido jauno jautājumu. Šie dati ietver:
 8. Lietotājvārds – lietotāja lietotājvārds – burtu teksts ar izmēru līdz 40 rakstzīmēm. (Piem. “Mets”)
 9. Nosaukums – jautājuma nosaukums - burtu teksts ar izmēru līdz 50 rakstzīmēm. (Piem. “Kāpēc php un Javascript??”)
 10. Teksts – jautājumu teksts – burtu teksts ar izmēru līdz 200 rakstzīmēm. (Piem. “teksts”)
11. Nosaukums un teksts tiks ievadīti no klaviatūras. Lietotājvārds un izveides datums ievadīti automātiski, kad lietotājs iesniegs jautājumu
12. **Komentāra pie jautājuma dati** tiks ievadīti sistēmā, kad lietotājs izveido jauno komentāru zem kādā jautājuma. Šie dati ietver:
 13. **Lietotājvārds** – lietotāja lietotājvārds – burtu teksts ar izmēru līdz 40 rakstzīmēm. (Piem. “Metss”)
 14. **Teksts** – jautājumu teksts – burtu teksts ar izmēru līdz 200 rakstzīmēm. (Piem. “komentāra teksts”)
 15. **Teksts** tiks ievadīts no klaviatūras. Lietotājvārds ievadīts automātiski, kad lietotājs iesniegs komentāru.

16. **Nosaukums** – tēmas nosaukums – burtu teksts ar izmēru līdz 50 rakstzīmēm. (Piem. “topic nosaukums”)
17. **Apraksts** - tematas apraksts - burtu teksts ar izmēru līdz 255 rakstzīmēm. (Piem. “Java apraksts”)
18. **Nosaukums**, Apraksts un autors tiks ievadīts no klaviatūras. Lietotājvārds un Faila nosaukums ievadītas automātiski, kad lietotājs iesniegs failu.
19. **Kategorijas dati** tiks ievadīti sistēmā, kad administrators pievieno jaunu kategoriju. Šie dati ietver:
 20. **Nosaukums** – kategorijas nosaukums – burtu teksts ar izmēru līdz 40 rakstzīmēm. (Piem. “Java”)
 21. **Apraksts** – kategorijas apraksts – burtu teksts ar izmēru līdz 255 rakstzīmēm. (Piem. “programmēšanas valoda”)
22. Teksts un novērtējums tiks ievadīti no klaviatūras.

2.2 Funkcionālās prasības

1. Reģistrēšana un atļauja:

- 1.1. Nodrošiniet lietotājam iespēju reģistrēties, ievadot nepieciešamo informāciju, piemēram, vārdu, e-pastu, paroli u. c. 1.2.
- 1.2. Ja obligātie lauki nav aizpildīti, ģenerējiet kļūdas ziņojumu.
- 1.3. Pārbaudiet ievadītā lietotājvārda vai e-pasta unikalitāti starp jau esošajiem lietotājiem

un atbilstības gadījumā izsniedziet attiecīgu paziņojumu.

1.4. Pēc veiksmīgas reģistrācijas saglabāt lietotāja datus datubāzē.

2. Lietotāja profils:

2.1. Nodrošināt iespēju autorizēties, izmantojot lietotājevārdu vai e-pastu kopā ar paroli.

2.2. Izvadiet kļūdas ziņojumu, ja autorizācijas laikā lauki ir tukši.

3. Forums:

3.1. Nodrošināt iespēju uzdot jautājumus, atstāt komentārus.

3.2. Sniegt lietotājiem tiesības veidot blogus un aptāujus.

3.3. Administrātoram nodrošināt iespēju veidot kategorijas, dzēst vestuļus.

4. Profils un rediģēšana:

4.1. Ļaujiet lietotājiem apskatīt un rediģēt savu profilu, tostarp mainīt paroli un atjaunināt informāciju.

4.2. Ļaujiet lietotājiem apskatīt citu lietotāju profilu.

5. Datu drošība:

5.1. Nodrošiniet lietotāju datus, tos šifrējot, lai novērstu nesankcionētu piekļuvi.

5.2. Nodrošiniet lietotājiem iespēju nepieciešamības gadījumā mainīt paroli.

6. Kategorijas:

6.1. Sniedziet lietotājiem iespēju apskatīt kategorijas.

6.2. Administrātors var veidot jaunus kategorijas.

2.3 Nefunkcionālās prasības

- Sistēmas saskanēs valodai jābūt pieejamai 1 valodā: angļu valodā.
- Tīmekļa lietojumprogrammai ir jāspēj pielāgoties mūsdienās izmantotajiem ekrāna izmēriem, lai to varētu izmantot uz dažāda izmēra monitoriem.
- Dizainam jābūt ieturētam organizācijas logotipa krāsās.
- Sistēmai jānodrošina lietotāju datu drošība, izmantojot šifrēšanas tehnoloģijas.

- Reģistrācijas procesam nevajadzētu aizņemt vairāk par 5 minūtēm (neskaitot administratora apstiprinājumu).
- Sistēmas saskarei jābūt viegli saprotamai un lietojamai.
- Sistēmas kļūdu paziņojumiem jābūt skaidriem un saprotamiem.
- Sistēmas reakcijas laikam nevajadzētu pārsniegt 1 sekundi.
- Sistēmai jābūt pieejamai 24 stundas diennaktī 7 dienas nedēļā.
- Sistēmai jābūt noturīgai pret pārslodzēm un kļūdām.
- Sistēmai jābūt atjauninātai un aprīkotai ar jaunākajām drošības funkcijām un uzlabojumiem.

Sistēmas ekrānu skices:

- Sistēmas reģistrācijas skice (skat. 2.3.1. att. un 2.3.2. att)

The diagram illustrates a registration form interface within a rectangular frame. It contains four input fields: a rectangular box labeled 'Username', a rectangular box labeled 'Password', a rounded rectangular button labeled 'Login', and a rounded rectangular button labeled 'Sign up'. The 'Username' and 'Password' fields are positioned side-by-side at the top. The 'Login' button is to the right of the 'Password' field. The 'Sign up' button is centered below the 'Username' and 'Password' fields.

2.3.1.att. Sistēmas reģistrācijas pieprasījuma nosūtīšana

Šī skice attēlo sistēmas interfeisu, kurā neregistrēti lietotāji var reģistrēties.

Username

Email

Password

Repeat Password

First name

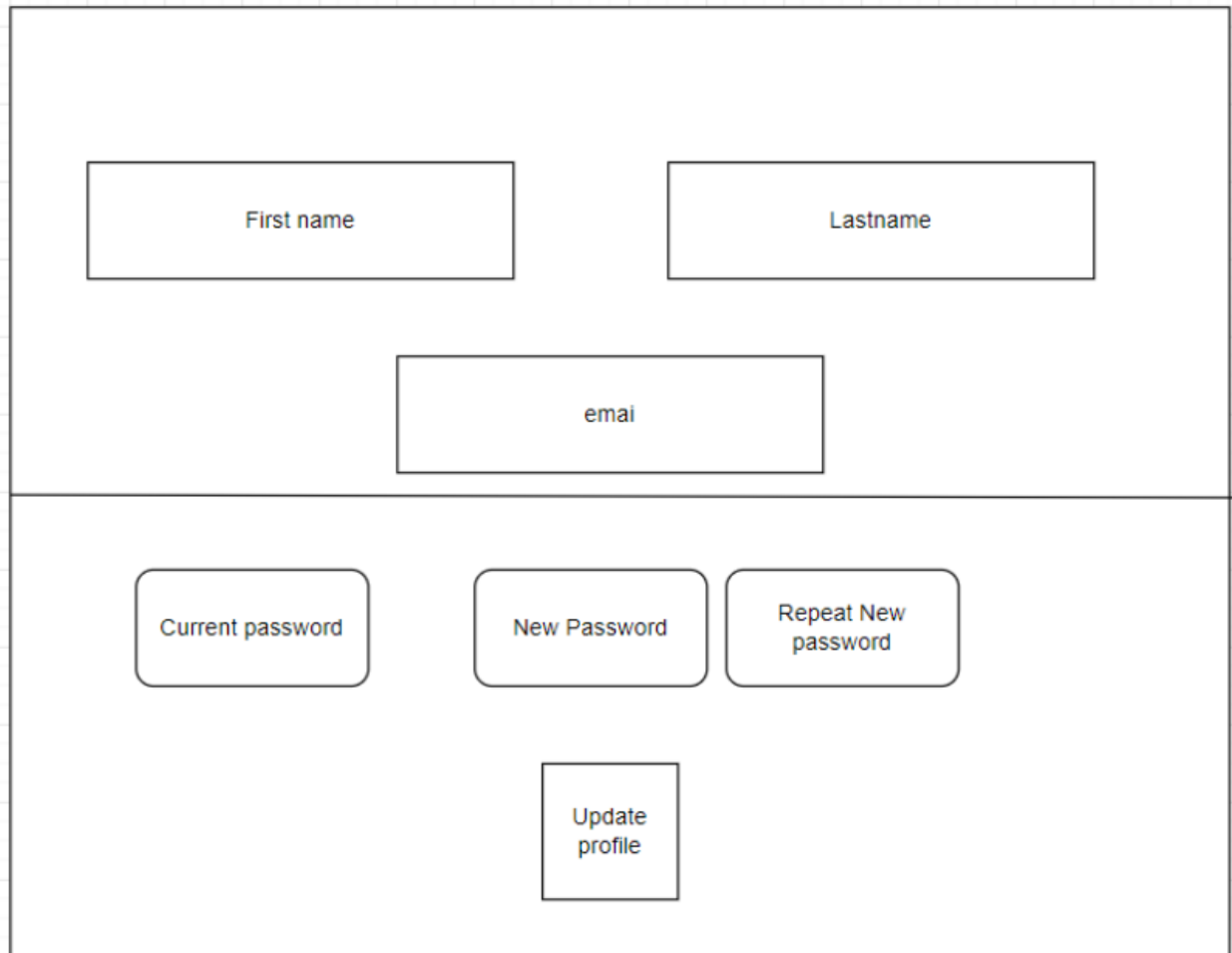
Lastname

Set Avatar

Signup

2.3.2.att. Sistēmas reģistrācijas pabeigšanas skice

Šī skice attēlo sistēmas interfeisu, kurā tiek parādīti lietotāji dati un to iespēja mainīt(parole , vārds,uzvārds, pasts , u.t.t)



The diagram illustrates a user profile update form, divided into two main sections by a horizontal line. The top section contains three input fields: 'First name' and 'Lastname' are side-by-side, and 'email' is centered below them. The bottom section contains three rounded rectangular input fields: 'Current password', 'New Password', and 'Repeat New password' are side-by-side, and an 'Update profile' button is centered below them.

First name		Lastname	
email			
Current password	New Password	Repeat New password	
Update profile			

2.3.3att. Lietotāju porfīla datus mainīšana.

3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS

Sistēma ir tiešsaistes platforma, kas radīta efektīvai darbībai pārlūkprogrammu vidē. Tā ir izveidota, sastāvot no divām pamata komponentēm: lietotāja interfeisa daļas (frontend) un

servera puses (backend).

Izstrādē tika izmantotas šādas programmēšanas vides:

- **Visual Studio 2019** ir moderna izstrādes vide, kas piedāvā izstrādātājiem ērtu un produktīvu darba vidi. Tā ietver integrētu kodu redaktoru, atklūdošanas rīkus, projektu un failu pārvaldību, kā arī daudzas citās funkcijas, kas palīdz veiksmīgi īstenot programmatūras projektus.
- **WAMPP:** WAMPP ir izstrādes vide, kas piedāvā Apache serveri, MySQL datubāzi, PHP interpretatorus. Tas nodrošina pilnvērtīgu izstrādes vidi web aplikāciju veidošanai.

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

- **HTML (versijā 5.3):** HTML ir hiperteksta marķēšanas valoda, ko izmanto tīmekļa lapu front-end daļas projektēšanai, izmantojot iezīmēšanas valodu. Tās funkcionalitāte ideāli piemērota sistēmas prasībām, piedāvājot iespēju strukturēt un formatēt saturu.
- **CSS (versijā 3):** CSS ir vienkārša izstrādāta valoda, kas paredzēta, lai vienkāršotu tīmekļa lapu izskatu veidošanas procesu. Tā ļauj piemērot stilus tīmekļa lapām, piedāvājot iespēju pielāgot izskatu un izvairīties no sarežģītas stilu iestatīšanas. Tās lietošanas ērtums ir galvenais faktors izvēlē.
- **Bootstrap:** Bootstrap ir tīmekļa izstrādes rīks, kas nodrošina gatavus stila šablonus un komponentus, lai vienkāršotu tīmekļa lapu izveidi. Tas ļauj izstrādātājiem viegli pielāgot tīmekļa lapas izskatu, izmantojot iebūvētos CSS klases un komponentus. Bootstrap piedāvā plašu funkciju klāstu, kas padara tīmekļa lapu izstrādi efektīvāku un ērtāku, īpaši jauniešiem izstrādātājiem. Šīs platformas lietošana nodrošina vienkāršu un konsistentu dizainu, kas ir būtisks faktors izvēlē, īpaši tiem, kuriem nav plašas pieredzes tīmekļa izstrādē.
- **JavaScript (versijā 1.8.5):** JavaScript ir slavena skriptu valoda, ko izmanto, lai padarītu vietnes interaktīvas un pievienotu dinamiskumu. To plaši izmanto, lai uzlabotu vietņu funkcionalitāti un piedāvātu lietotājiem dinamiskas un responsīvas pieredzes.

Servera daļa tiek veidota, izmantojot dažas valodas un rīkus, kas aprakstītas turpmāk:

- **MySQL (versijā 8.0):** MySQL ir relāciju datubāzes pārvaldības sistēma, kas tiek

izmantota datu glabāšanai un iegūšanai no citām programmatūras lietojumprogrammām. Tā ir izvēlēta kā datu bāzes risinājums, jo piedāvā efektīvu datu organizāciju, labu atbalstu transakcijām un ir labi pielāgojama Laravel lietojumprogrammu izstrādei.

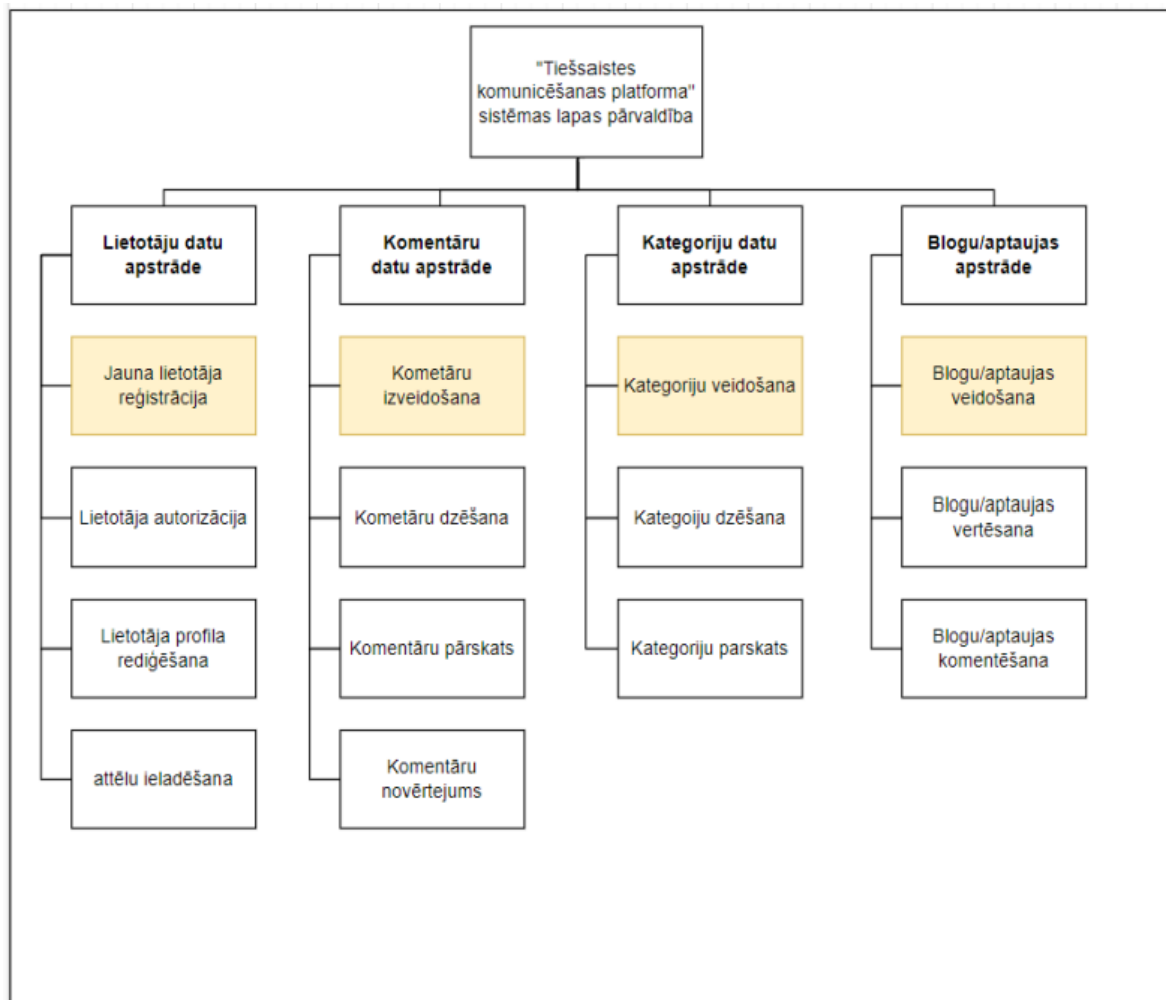
- **phpMyAdmin:** phpMyAdmin ir PHP rakstīta programma datubāžu pārvaldībai, kas piedāvā grafisku lietotāja saskarni MySQL datubāzēm. Tas atvieglo datubāžu pārvaldību un manipulāciju ar datiem.

4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

4.1 Sistēmas struktūras modelis

Sistēmai (skat. 4.1. att.) sastāvēs no četrām apakšsistēmām: lietotāju datu apstrādes,

komentāru datu apstrādes, kategoriju datu apstrādes un blogu/aptaujas datu apstrādes apakšsistēmām.



4.1. att. Sistēmas funkcionālā dekompozīcijas diagramma

- **Lietotāju datu apstrādes apakšsistēma** ir atbildīga par lietotāju reģistrāciju, autorizāciju, rediģēšanu un attēlu ielādēšanu. Šī apakšsistēma saglabā informāciju par lietotājiem, to datiem un citiem datiem.
- **Komentāru datu apstrādes apakšsistēma** ir atbildīga par lietotāju komentāru

izveidošanu un novērtēšanu. Šī apakšsistēma saglabā informāciju par lietotāju komentāriem, to datiem un citiem datiem.

- **Blogu/aptāņu datu apstrādes apakšsistēma** ir atbildīga par blogu/aptāņu veidošanu un mainīšanu. Šī apakšsistēma saglabā informāciju par blogu/aptāņu , to datiem un citiem datiem.
- **Kategoriju datu apstrādes apakšsistēma** ir atbildīga par kategoriju izveidošanu un dzēšanu . Šī apakšsistēma saglabā informāciju par kategoriju, to datiem.

4.2 Sistēmas ER modelis

Sistēmas ER-modelis sastāv no 4 entītijām (skat. 1. piel.), kas nodrošina pamat informācijas uzglabāšanu un apstrādi. Sistēmas ER-modeļa entītijas ir:

- **Lietotājs** – šī entītijā reprezentē sistēmas lietotājus. Katram lietotājam ir unikāls lietotājvārds, e-pasts un parole. Lietotājs var būt gan administrators, , gan parasts lietotājs(lietotājs).
- **Bilde** – šī entītijā parada ielādētu bildi, kas tiek lejup lādēti nolietotāja vai administratora puses.
- **Jautājums** – šī entītijā reprezentē foruma jautājumus. Katram jautājumam ir

unikāls reģistrācijas numurs, lietotāja vārds, nosaukums un teksts.

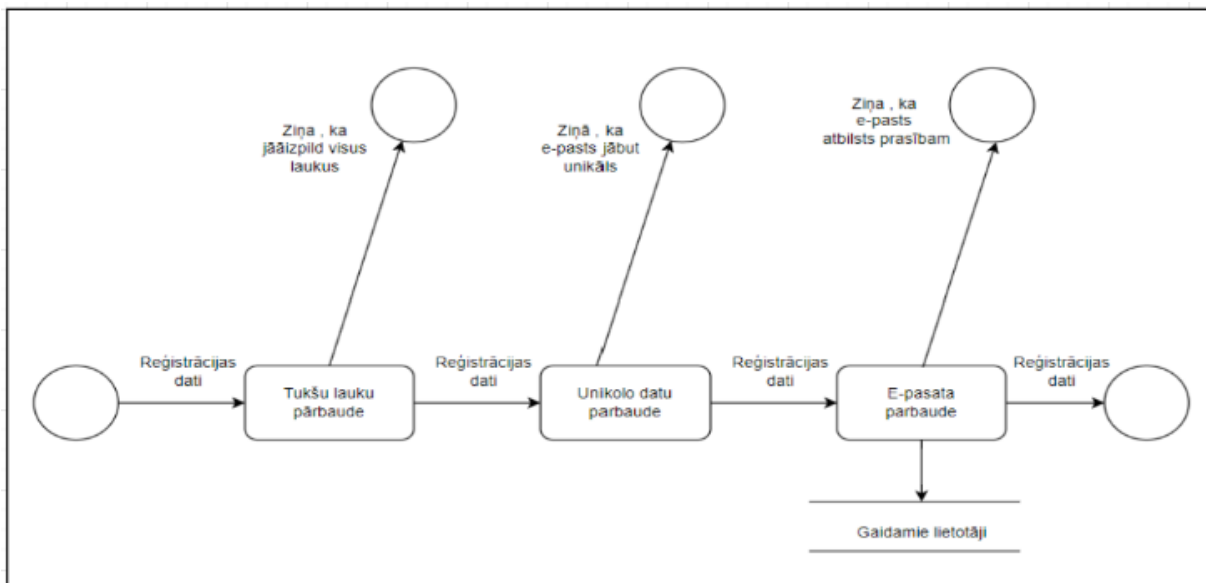
- **Jautājuma komentārs** – šī entītija reprezentē foruma komentārus. Katram komentāram ir unikāls reģistrācijas numurs, lietotāja vārds un teksts.
- **Kategorija** - šī entītija reprezentē kategorijas. Katram kategorijas ir nosaukums, apraksts un unikāls reģistrācijas numurs.
- **Jautājuma kategorija** - šī entītija reprezentē jautājuma kategorijas. Katram kategorijas ir nosaukums, apraksts un unikāls reģistrācijas numurs.
- **Lietotājs** ir saistīts ar **Jautājumiem**, jo viens lietotājs var veikt vairākas jautājumus.
- **Jautājums** ir saistīts ar **Komentāriem** un **Lietotājiem**, jo katrs komentārs ir saistīts ar konkrētu jautājumu un lietotāju.
- **Lietotājs** ir saistīts ar **Failiem**, jo viens lietotājs var lejup lādēt ne tikai bildi profilā, bet arī kā fonu aptaujā / blogā.

4.3 Funkcionālais sistēmas modelis

- *Datu plūsmu modelis*

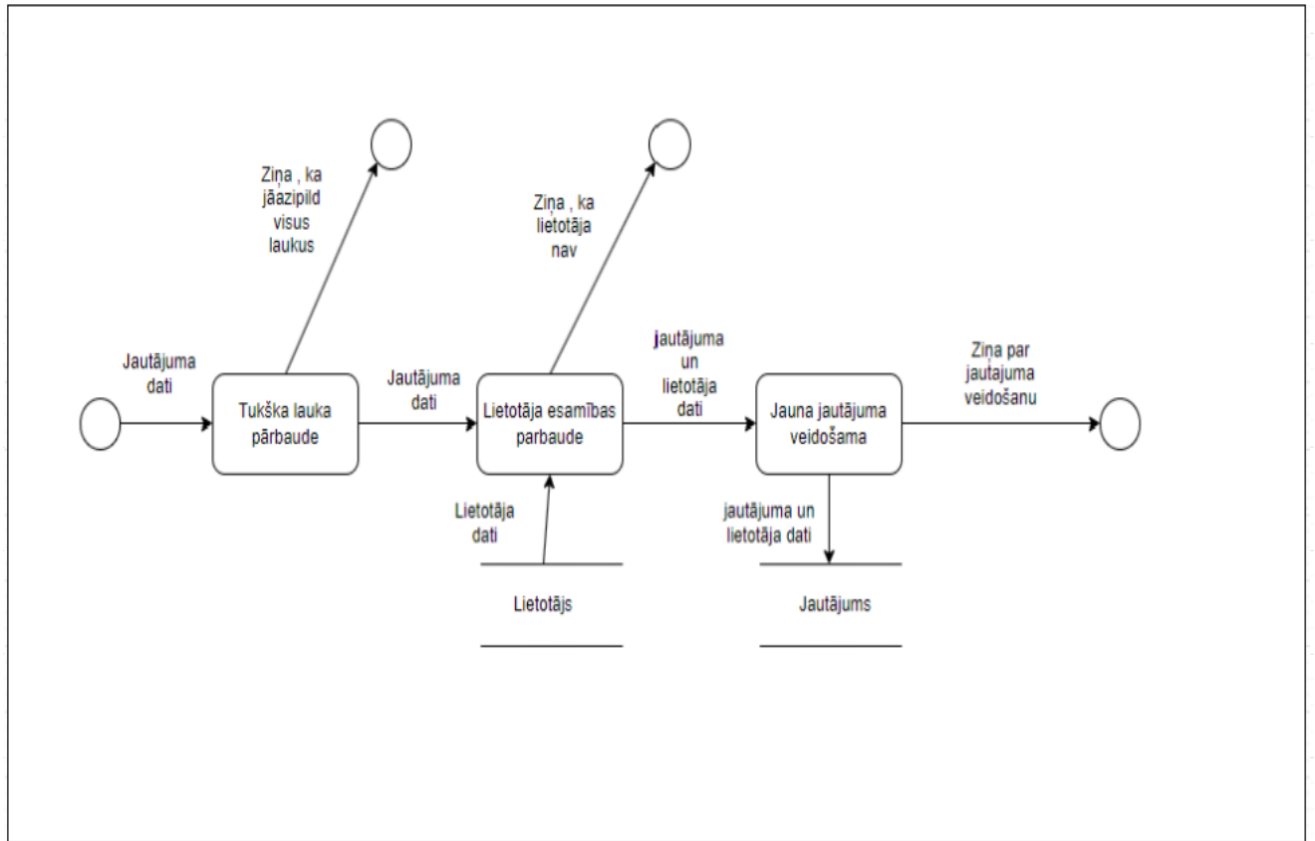
Datu apstrādes procesi:

4.3.1 Jauna lietotāja reģistrācija – šīs operācijas procesā (skat. 4.3.1.) lietotājam ir jāievada dati, kuras, pēc veiksmīgu validāciju tiek ievadītas datubāzē. Kad reģistrācija ir pabeigta veiksmīgi lietotājs var ielogoties un aiziet vietnē.



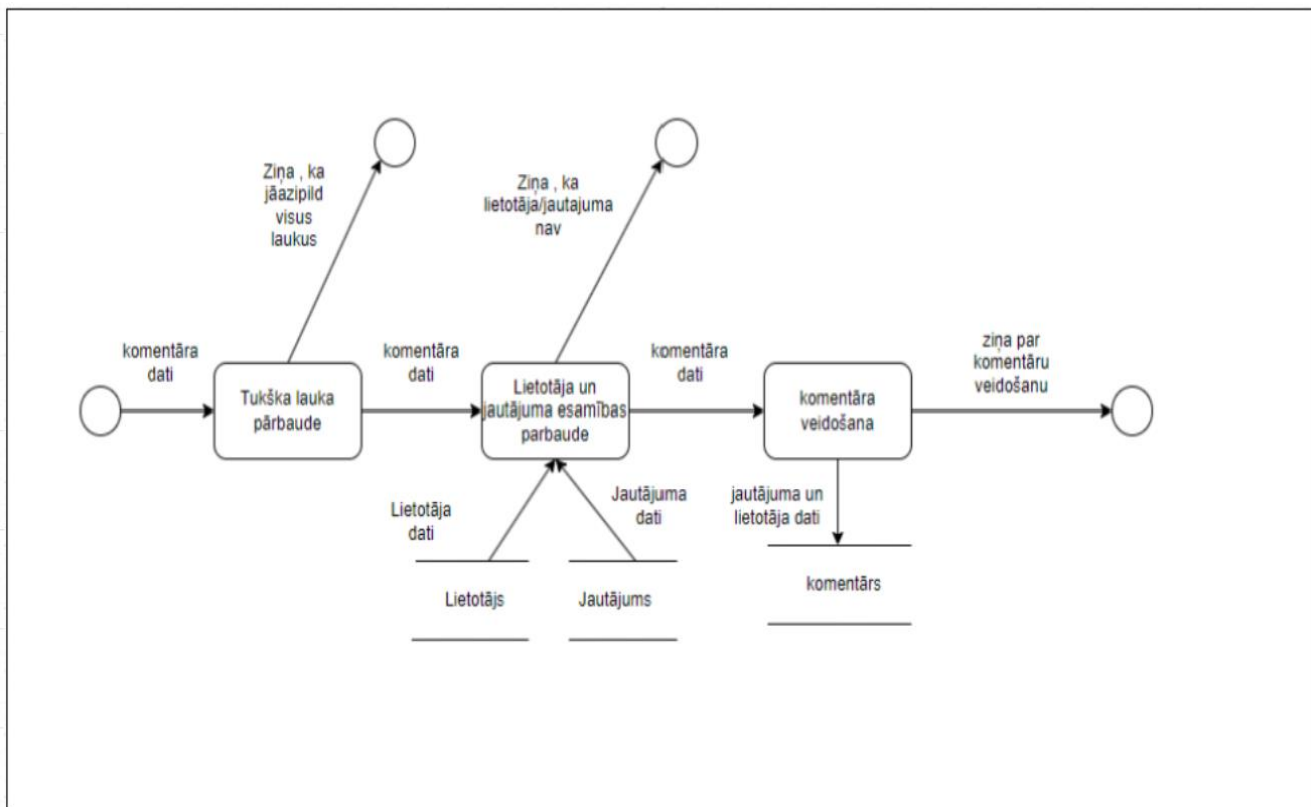
4.3.1. att. Jauna lietotāja reģistrācija plūsmu diagramma.

- **Jautājuma izveidošana** – šīs operācijas procesā (skat.4.3.2. att.) lietotājam ir jāievada dati, kuras, pēc veiksmīgu validāciju, tiek ievadītas datubāzē un tiks parādīts teksts ar veiksmīgu jautājuma veidošanu.



4.3.2.att. Jautājuma izveidošana plūsmu diagramma

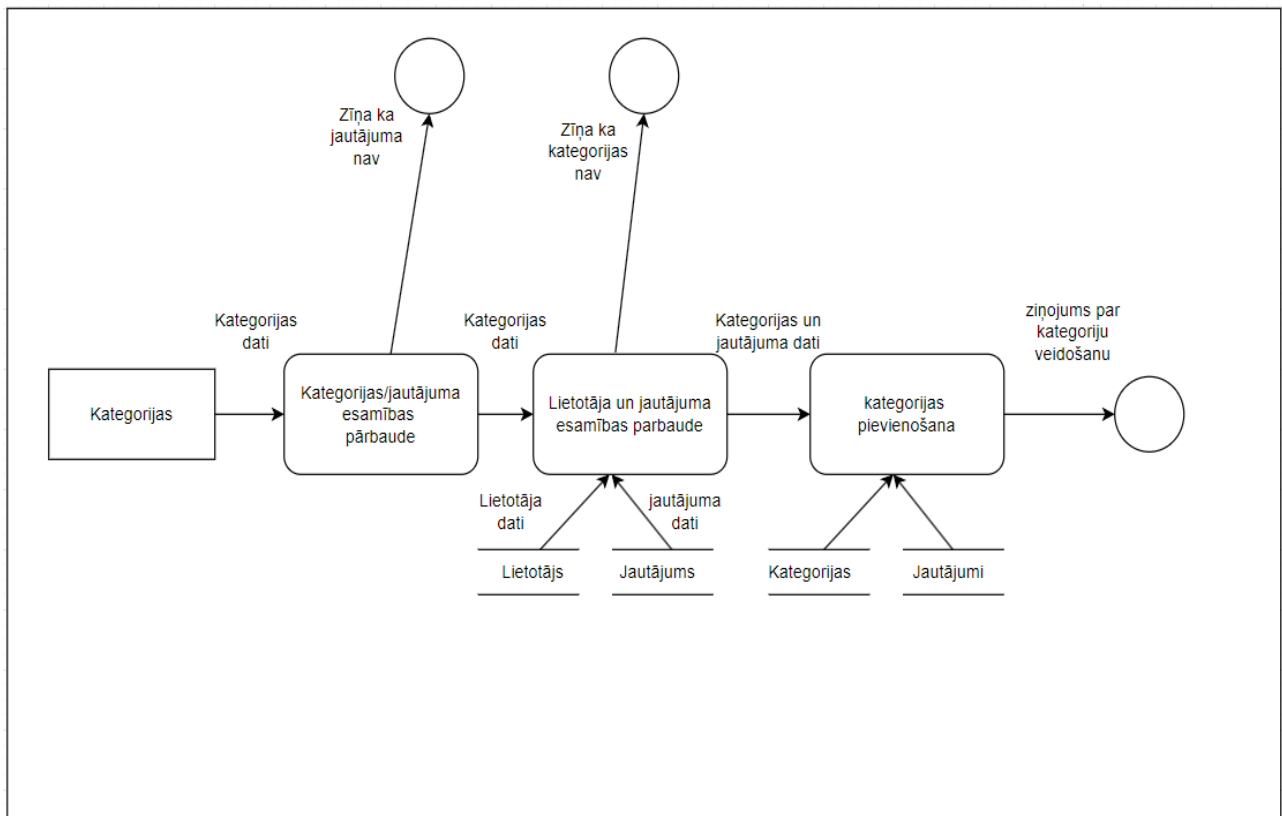
- **Komentāru izveidošana** - šīs operācijas procesā (skat. 4.3.3. att.) lietotājam ir jāievada dati, kuras, pēc veiksmīgu validāciju, tiek ievadītas datubāzē un tiks parādīts teksts ar veiksmīgu komentāra veidošanu.



4.3.3.att. Jautājuma izveidošana plūsmu diagramma.

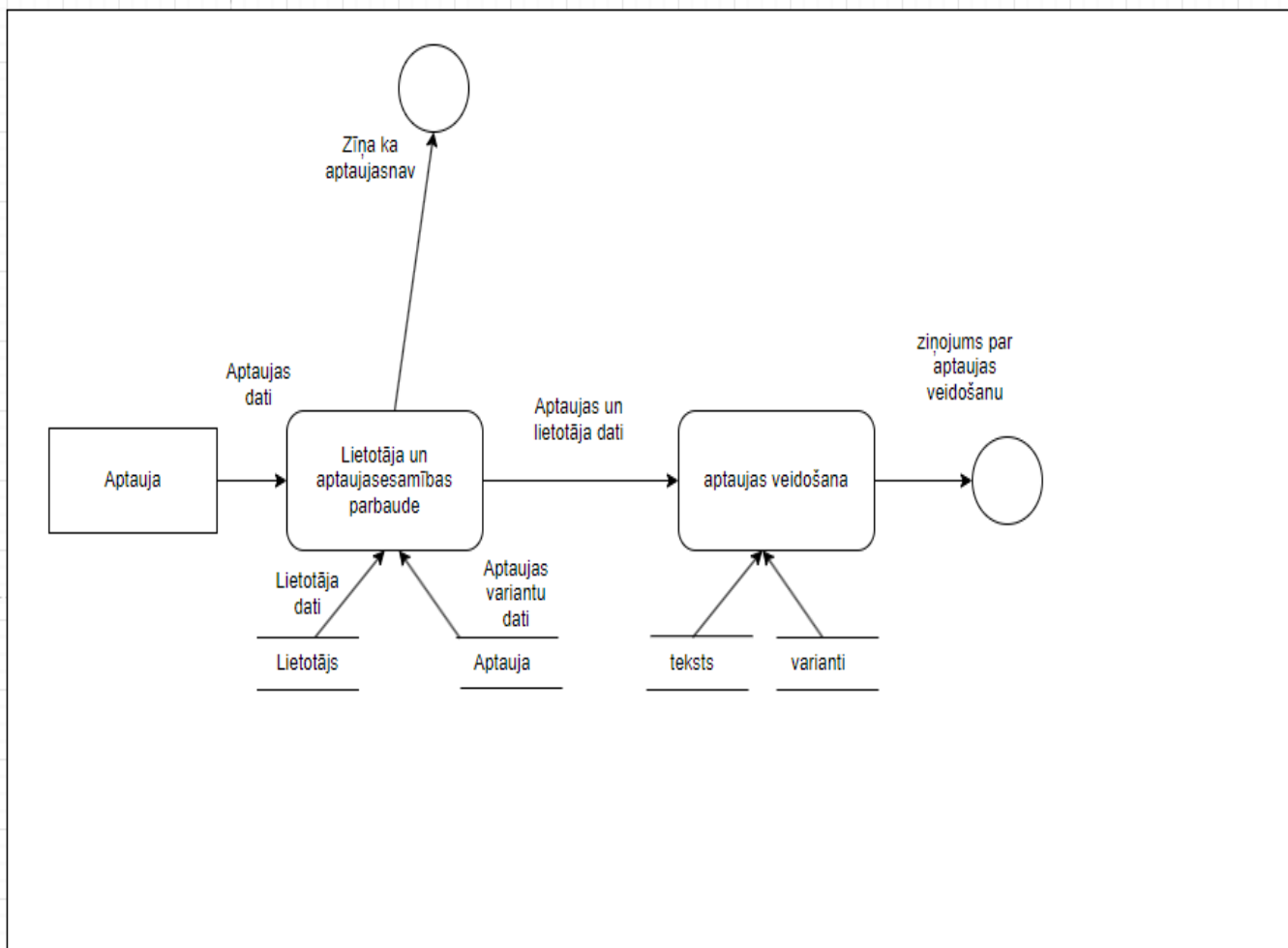
- **Kategoriju izveidošana** - šīs operācijas procesā (skat. 4.3.4. att.)

administratoram ir jāievada dati, kuras, pēc veiksmīgu validāciju, tiek ievadītas datubāzē un tiks parādīts teksts ar veiksmīgu kategoriju izveidošanu.



4.3.4.att. Kategorijas izveidošana plūsmu diagramma.

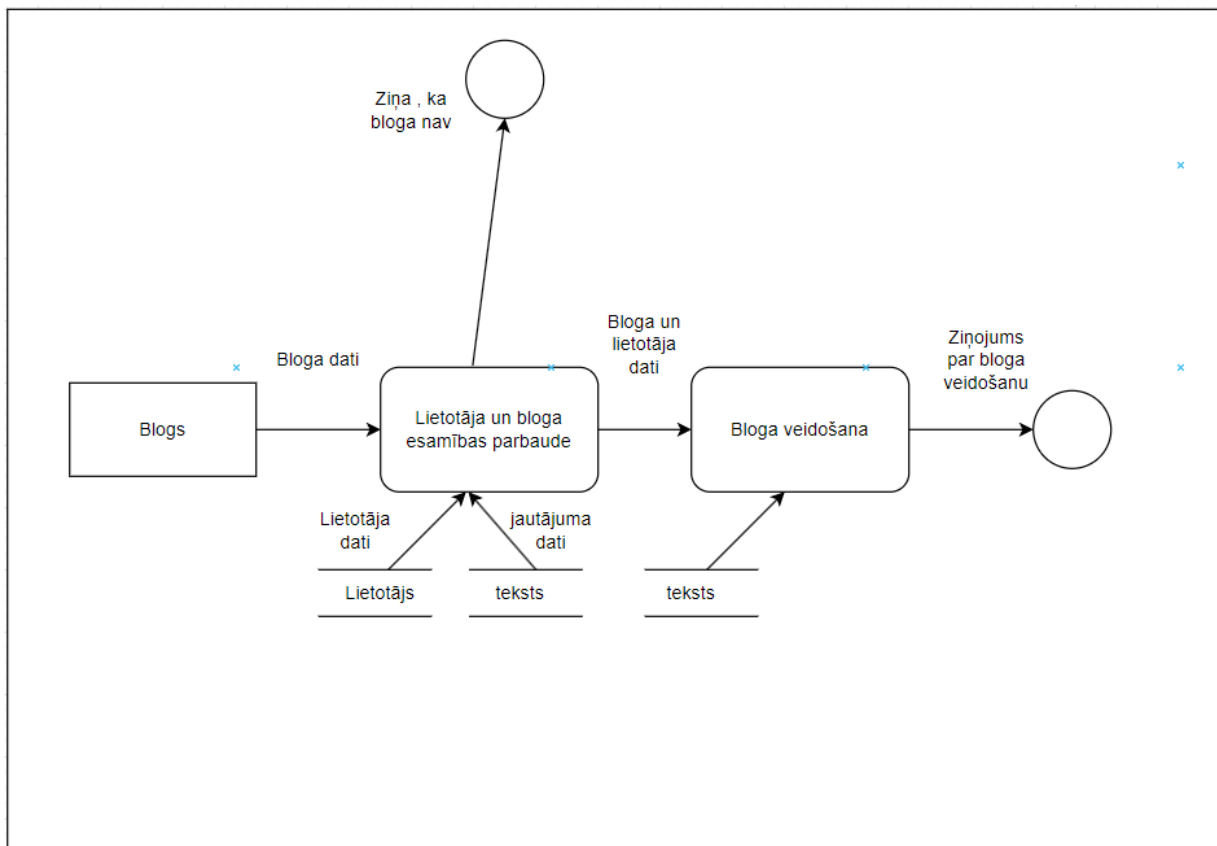
Aptāņu veidošana - šīs operācijas procesā (skat. 4.3.5. att.) lietotājam ir jāievada dati, kuras, pēc veiksmīgu validāciju, tiek ievadītas datubāzē un tiks parādīts teksts ar veiksmīgu aptāņu izveidošanu.



4.3.5.att. Aptaujas izveidošana plūsmu diagramma.

Bloga veidošana - šīs operācijas procesā (skat.4.3.6. att.) lietotājam ir jāievada dati, kuras, pēc veiksmīgu validāciju, tiek ievadītas datubāzē un tiks parādīts teksts ar veiksmīgu aptauju izveidošanu. (Blogs tik izveidots lai lietotājs var dalīties ar kadiem ziņiem bez

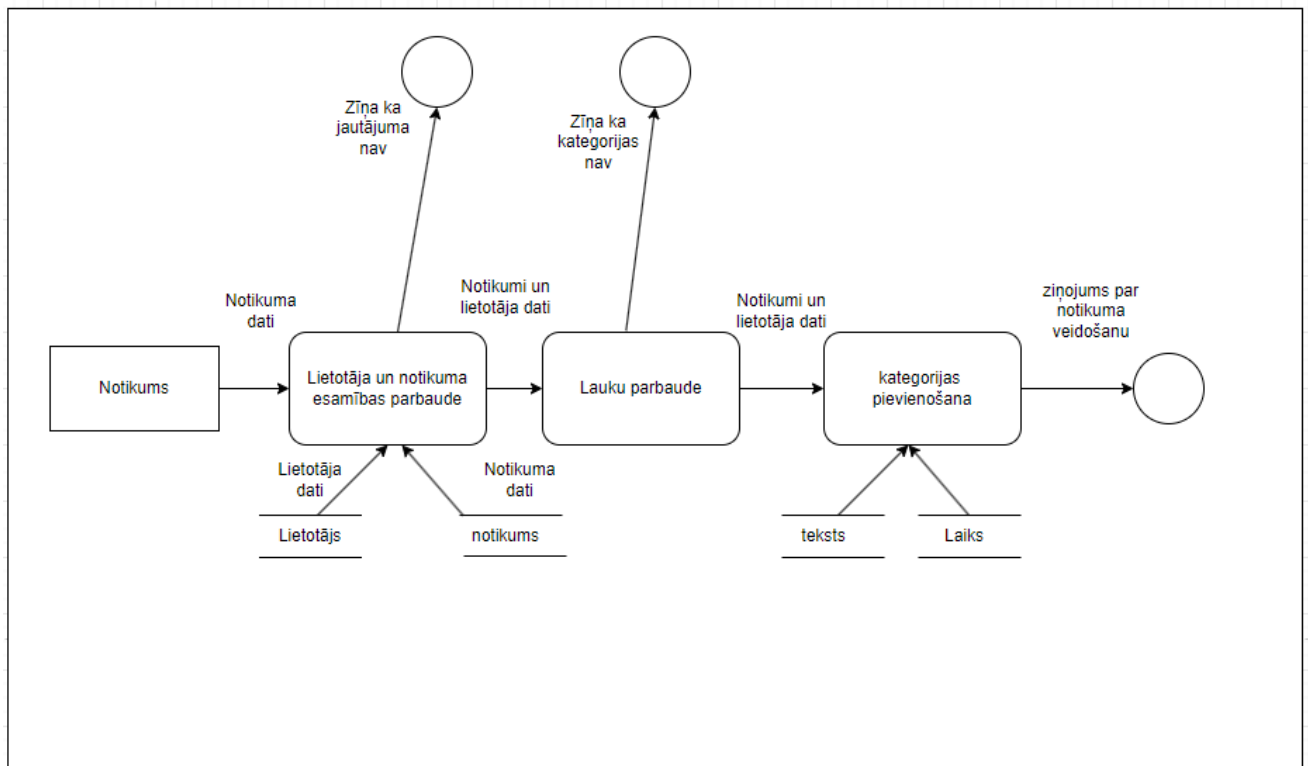
atbildem).



4.3.6.att. Bloga izveidošana plūsmu diagramma.

Notikuma veidošana - šīs operācijas procesā (skat. 4.3.7. att.) lietotājam ir jāievada dati, kuras, pēc veiksmīgu validāciju, tiek ievadītas datubāzē un tiks parādīts teksts ar veiksmīgu aptauju izveidošanu. (Notikums izveidots lai parādītu

kadu specifisku dati un notikumu kas tik atbilst tam datumam).



4.3.7.att. Notikuma izveidošana plūsmu diagramma.

5. DATU STRUKTŪRU APRAKSTS

Veidojot datubāzi, liela uzmanība tika pievērsta tabulām, lai nodrošinātu efektīvas attiecības un vislabāko iespējamo datu glabāšanas vidi. Datu bāze sastāv no - tabulām, kas satur sevī informāciju par gaidāmiem lietotājiem, lietotājiem, kategorijām, lietotāju jautājumiem un komentāriem(forums).

- Tabula “**users**” ietver 11 lauki un glabā datus par lietotājus.

- Tabula **Topics** ietver 5 laukus un glabā datus par izveidotam tēmatam.
- Tabula **Posts** ietver 6 lauki un glabā datus par pašiem jautajumiem iekš postā.
- Tabula **Post Votes** ietver 5 lauki un glabā datus par komentaru reitingu.
- Tabula **“Blogs”** ietver 6 lauki un glabā datus par blogiem.
- Tabula **“Blog votes”** ietver 5 lauki un glabā datus balsiem blogā.
- Tabula **“Events”** ietver 6 lauki un glabā datus par notikumiem.
- Tabula **“Event_info”** ietver 5 lauki un glabā tekstu par notikumiem.
- Tabula **“Polls”** ietver 8 lauki un glabā datus par aptauju.
- Tabula **“Poll_options”** ietver 5 lauki un glabā aptaujas konfiguraciju.
- Tabula **“Poll_votes”** ietver 4 lauki un glabā datus kurš tik balso.
- Tabula **“Categories”** ietver 3 lauki un glabā datus par kategorijām un datiem.
- Tabula **“Pwdreset”** ietver 5 lauki un glabā datus par parola mainīšanu.

1. Tabula **“users”** satur informāciju par lietotājiem, kuri jau tiek izveidoti sistēmā. Tie tiek automātiski izveidoti, kad lietotājs tiek veiksmīgi reģistrējies. Tajā ietilpst dati, piemēram, lietotāja id, vārds, uzvārds, e-pasts, dzimusi, avatāru, lietotājvārds, parole, apraksts, lauks ar pārbaudīšanu vai lietotājs ir administrators.

1.tabula

Tabulas “**users**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	IdUsers	int	11	Unikālais kods
2.	UserLevel	int	5	Vai lietotājs ir admins
3.	f_name	varchar	50	lietotāja vārds
4.	l_name	varchar	50	lietotāja uzvārds
5.	uidUsers	tinytext	20	Lietotāja segvārds
6.	emailUsers	tinytext	20	Lietotāja pasts
7.	pwdUsers	longtext	50	Lietotāja parole
8.	gender	char	1	Lietotāja dzimums
9.	Headline	varchar	500	apraksts zem profila
10.	bio	varchar	4000	lietotāja apraksts
11.	IserImg	varchar	500	lietotāja bilde

2. Tabula “**Topics**” satur informāciju par tematam kurie uzraksta lietotāji , parada , kad tik izveidots un kurš lietotājs izveido.Tajā ietilpst dati , piemeram , datums , tema rūnāšanai , kategorija.

2.tabula

Tabulas “**Topics**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
			rs	

1.	topic_id	int	11	Unikālais kods
2.	topic_subject	varchar	255	Vai lietotājs ir admins
3.	topic_date	datetim e	10	lietotāja vārds
4.	topic_cat	int	255	lietotāja uzvārds
5.	topic_by	int	255	kurš lietotājs uzrakstīja

3. Tabula “**Posts**” satur informāciju par komentariem kurie raksto lietotāji iekš topikā.
Tajā ietilpst dati, piemērām, komentarija balsi , komentara datums ,komentara teksts, autors.

3.tabula

Tabulas “**Posts**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	post_id	int	11	Unikālais kods

2.	post_content	text	255	komentars
3.	post_date	datetime	10	kad lietotājs uzrakstīja
4.	post_topic	int	255	kura topikā ir komentars
5.	post_by	int	255	Lietotājs kurš uzraksta
6.	post votes	int	-	komentarija reitings

4. Tabula “**Post_votes**” satur informāciju par balsojumiem , kurie atbilst kads reitings būs komentaram. Tajā ietilpst dati, piemēram, balsa datums , balsa autors.

4.tabula

Tabulas “**Post_votes**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	vote_id	int	11	Unikālais kods
2.	votePost	int	255	kurā

				komentārā bija balsojums
3.	voteBy	int	255	kurš lietotājs balso
4.	voteDate	date	10	datums balsojumam
5.	vote	int	255	+1/-1 balsojums

5. Tabula “**Blogs**” satur informāciju par blogu kuru veido lietotājs , un iekšdatus. Tajā ietilpst dati, piemērām, bloga nosaukums , bilde , datums , balsu daudzums , iekšteksts.

5.tabula

Tabulas “**Blogs**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	blog_id	int	11	Unikālais kods
2.	blog_title	varchar	100	bloga nosaukums

3.	blog_img	varchar	1000	bloga bilde
4.	blog_by	int	50	lietotājs kurš veido blogu
5.	blog_date	date	10	bloga datums
6.	blog_votes	int	255	bloga balsojumi
7.	blog_content	longtext	5000	teksts iekš blogā

6. Tabula “**Blog votes**” satur cik balsojumus ir blogam.

Tajā ietilpst dati, piemērām, balsa datums , kurš balso.

6.tabula

Tabulas “**Blog votes**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	vote_id	int	11	Unikālais kods
2.	voteBlog	int	255	kurā bloga tik veikts balsojums
3.	voteBy	int	255	kurš lietotājs

				balsoja
4.	voteDate	date	10	balsojuma datums
5.	vote	int	255	balsojums

7. Tabula “**Events**” satur informāciju par notikumu , to iekšinformāciju un datumu , kad būtu notikums. Tajā ietilpst dati, piemēram, notikuma datums , nosaukums , bilde , autors.

7.tabula

Tabulas “**Events**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	event_id	int	11	Unikālais kods
2.	event_by	int	50	notikuma

				autors
3.	title	varchar	1000	nosaukums
4.	date_created	date	10	kada laika izveidots
5.	event_date	varchar	10	notikuma datums
6.	event_image	varchar	255	notikuma bilde

8. Tabula “**Event_info**” satur informāciju par notikumu , to iekšinformāciju un datumu , kad būtu notikums. Tajā ietilpst dati, piemēram, apraksts , nosaukums.

8.tabula

Tabulas “**Event_info**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	event_id	int	11	Unikālais kods
2.	event	int	50	notikuma id
3.	title	varchar	100	nosaukums

4.	headline	varchar	100	apgalvojums
5.	description	varchar	6000	teksts

9. Tabula “**polls**” satur informāciju par aptauju un to izvēli.

Tajā ietilpst dati, piemēram, aptaujas variants ,varianti , autors.

9.tabula

Tabulas “**polls**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	Id	int	10	unikāls identifikators
2.	subject	varchar	255	aptāujas nosaukums

3.	created	datetime	10	laiks kad izveidots
4.	modified	datetime	10	laiks kad mainīts
5.	status	enum	1	vai ir izveidots
6.	created_by	int	50	kurš izveidoja
7.	poll_desc	varchar	5000	apraksts
8.	locked	int	50	atvērts/aizvērt s

10. Tabula “**poll_options**” satur informāciju par aptauju , un visus izveidotus variantus.
(tas nosaukums un unikālais id).

10.tabula

Tabulas “**poll_options**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	Id	int	10	unikāls identifikators
2.	poll_id	int	50	aptauja identifikators(kur ir

				izvēle)
3.	name	varchar	255	variantu nosaukums
4.	created	datetime	10	kad izveidots
5.	modified	datetime	10	kad izmainits
6.	status	enum	1	status vai ir izveidots

11. Tabula “**poll_votes**” satur informāciju par balsošanu aptaujā.

11.tabula

Tabulas “**poll_votes**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	Id	int	10	unikāls identifikators
2.	poll_id	int	50	izvēlētie varianti
3.	poll_option_id	int	50	unikāls identifikators

4.	vote_by	int	50	kurš lietotājs balso
----	---------	-----	----	-------------------------

12. Tabula “**Categories**” satur informāciju par kategorijām.
Tajā ietilpst dati, piemēram, kategorijas nosaukums un apraksts.

12.tabula

Tabulas “**Categories**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	cat_id	int	10	unikāls identifikators
2.	cat_name	varchar	255	kategorijas nosaukums
3.	cat_description	varchar	255	apraksts

13. Tabula “**pwdreset**” satur informāciju par paroli un to mainīšanu

13.tabula

Tabulas “**pwdreset**” struktūra

Nr.	Nosaukums	Tips	Izmērs	Apraksts
1.	pwdResetID	int	10	unikāls identifikators
2.	pwdResetEmail	text	255	pastas mainīšana
3.	pwdREsetSelecto	text	255	paroles

	r			mainīšana
4.	pwdResetToken	longtext	255	paroles mainīšana
5.	pwdResetExpires	text	255	paroles mainīšana kļūda

6. LIETOTĀJA CEĻVEDIS

6.1 Sistēmas prasības aparatūrai un programmatūrai

Neskatoties uz to, ka sistēma ir realizētā ar WEB-tehnoģiju palīdzību un neprasa specifiskas ierīces, ir daži prasījumi stabilai darbībai:

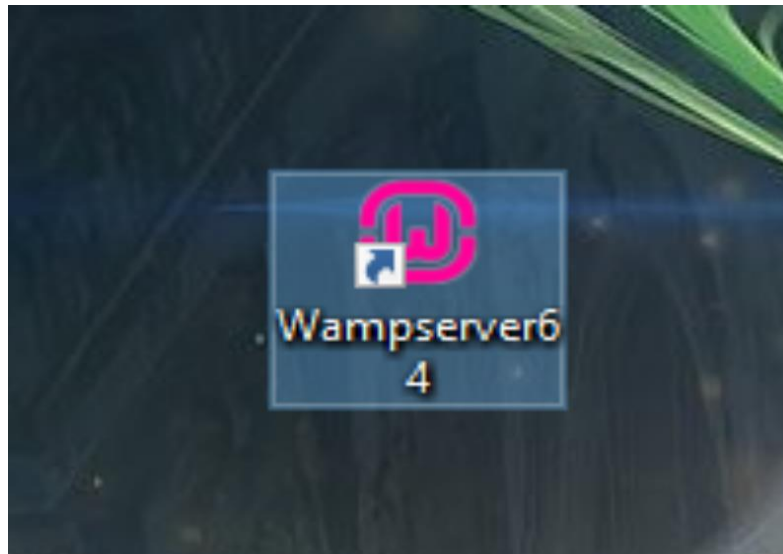
- Windows operētājsistēma.
- Koda redaktors(piemēram: Visual Studio Code)
- PHP bāzēts rīks Laravel
- Stiprs un stabils interneta signāls.
- RAM datora atmiņa vismaz 2 Gb.
- Web-aplikācijai ir jāstrādā no visiem interneta pārlūkiem, bet labāk ņemt kādu no pazīstamākajiem, piemēram: Opera, Edge, Firefox utt.

6.2 Sistēmas instalācija un palaišana

Sakarā ar to, ka programma ir realizēta ar WEB-tehnoloģijas palīdzību, lai to palaist ir jāveic sekojošus soļus:

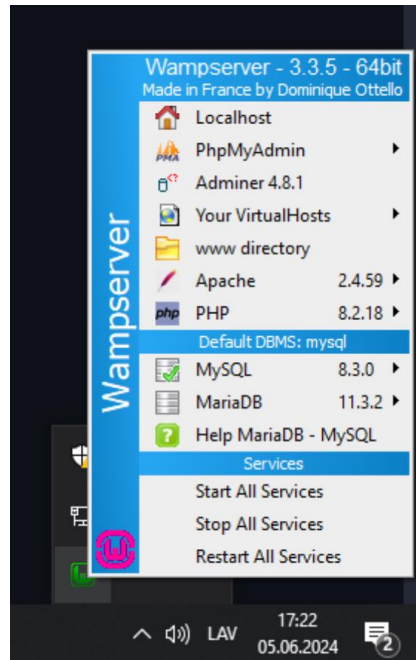
- Uzinstalēt uz datoru serveri, ar datubāzi un iespēju uztaisīt virtualo hostu(piemēram: Wampp)
- Uzinstalēt uz datoru PHP programmēšanas valodu(nāc iekšā ar Wampp)
- Uzinstalēt Composer - lietojumprogrammas līmeņa atkarību pārvaldnieks PHP programmēšanas valodai.
- Lejupielādēt koda mapi

(https://github.com/rvt-prog-kval-24/programm-taju_soci-lais_tikls) un ielikt zem servera mapes C/: diskā htdocs (ceļš C:\wampp\www).



6.1. attēls - wamp icon.

- Pirmokārt , jāatvert Wamp programmu(PhpMyadmin un Apache tik atvēras automatiski.) 6.1. att.



6.2. attēls - wamp menu.

Pēc tam aiziem uz windows menu un tur nepieciešams nospiegt uz **wamp** icon un pēc tam uz **PhpMyAdmin**.(6.3. att)



phpMyAdmin
Welcome to phpMyAdmin

Language

English

Log In

Username: root

Password:

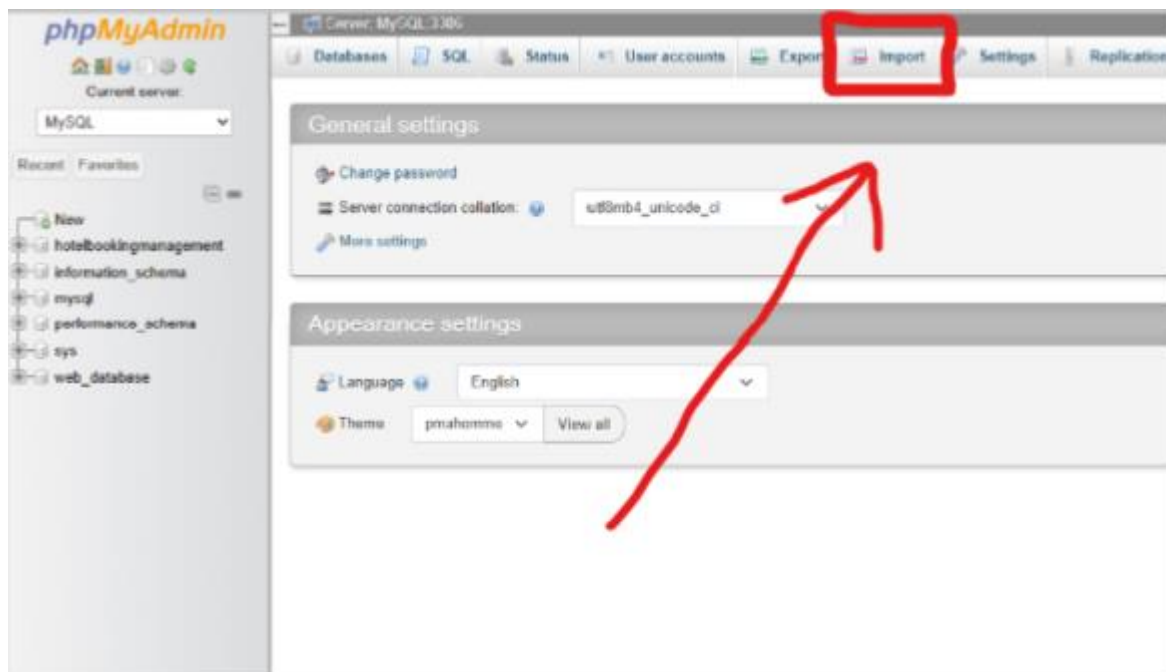
Server choice: MySQL

Log in

6.3. attēls - phpMyAdmin menu.

Iekš PhpMyAdmin laukiem rakstām - username: root , paroli nerakstām. (6.3. att.)

Jānospiež pogu **Import** (6.4. att.)



6.4. attēls - importēšana.

Jāimporte failu - **web_database.sql** (6.5. att)

File to import:

File may be compressed (gzip, zip) or uncompressed.
A compressed file's name must end in `[format].[compression]`. Example: `.sql.zip`

Browse your computer: (Max: 128MiB)

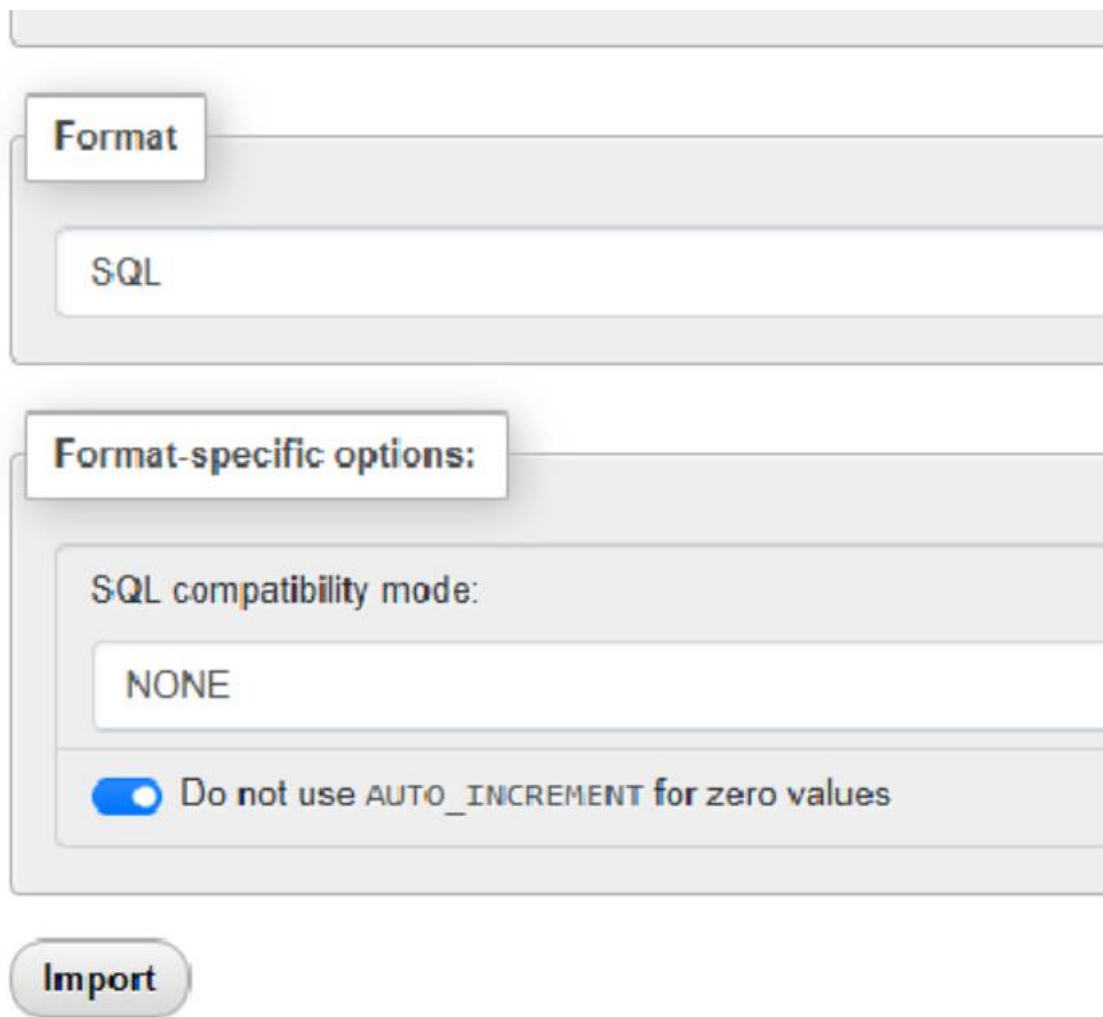
Upload file:

You may also drag and drop a file on any page.

Character set of the file:

6.5 attēls - datu bāzes importēšana.

Jaiesniedz pogu - **Import** un aizliegt phpMyAdmin (6.6. att)



The screenshot shows the 'Import' interface in phpMyAdmin. At the top, there is a 'Format' dropdown menu currently set to 'SQL'. Below this, under the heading 'Format-specific options:', there is a 'SQL compatibility mode:' dropdown set to 'NONE'. A toggle switch for 'Do not use AUTO_INCREMENT for zero values' is turned on. At the bottom of the form is a large 'Import' button.

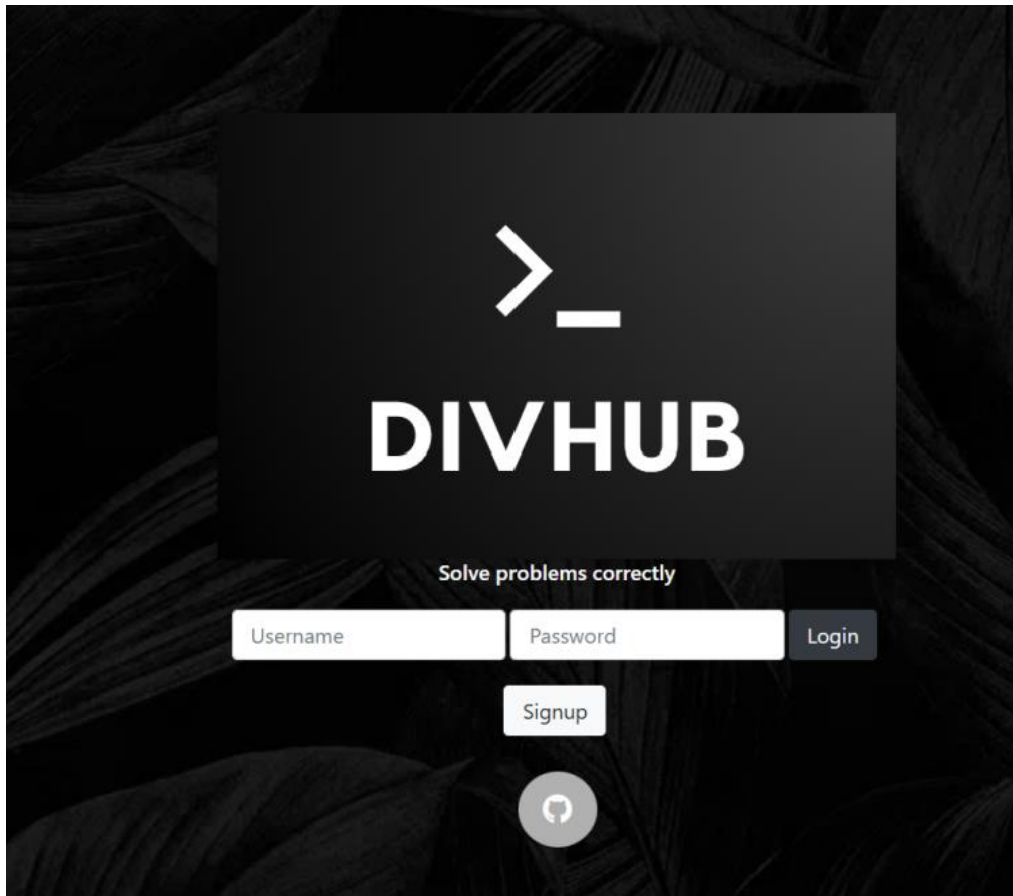
6.6 attēls - Import Poga.

- Pieslēgties jebkurai interneta tīklam vai izmantot mobilo datus.
- Izvēlēties pārlūku, ar kuru palīdzību atvērt pašu sistēmu.
- Un ierakstīt linku: localhost/website

7. Programmas apraksts

- *Autorizācija reģistrētiem lietotājiem*
- Atvēram pārlūkā saitī: <http://localhost/website/login.php>

- Pirmokart paradas login lappuse , ir divi lauki - sēgvardām un parolei.Arī divas pogas – login/signup. Registracijai nospiežam pogu - **Signup** (7.1 att.)



5.1 attēls - login lappaspuse.

Reģistrācijas lappa - Jāievada datus lai pieteikties. Obligatie dati - slēgvards , pasts , parole.(7.2. att)

Signup and Begin!

Abcd pasts123@gmail.com

Password parole1234

Optional

First Name Vards Last Name Uzvars

Avatar

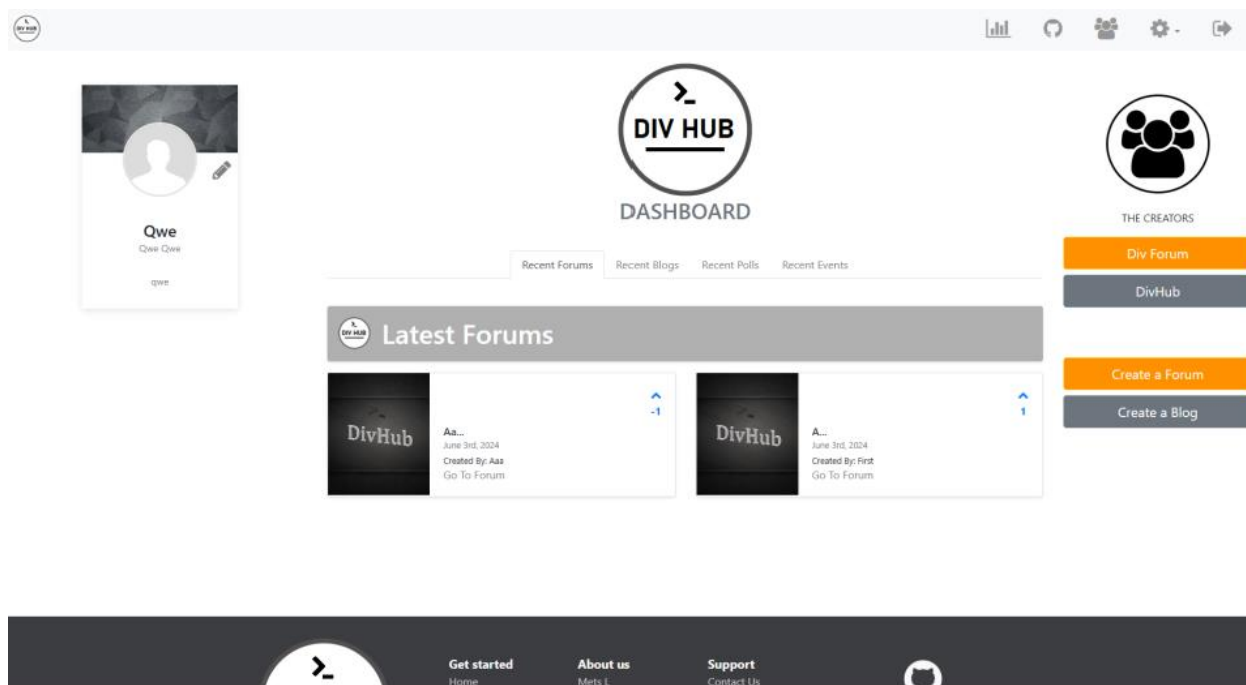
Set Avatar

Headline never give up!

ruby programmer

5.2. attēls - reģistrācija.

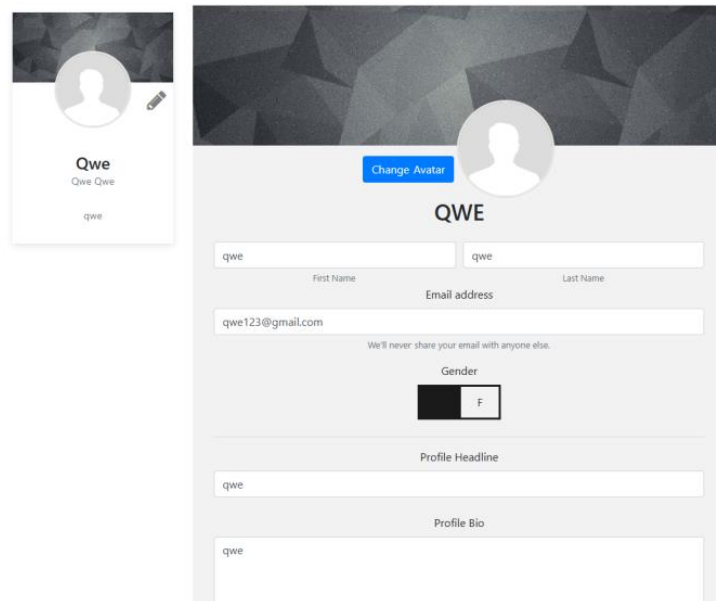
Galvena sociāla tīkla lappa kur ir viss vajadzīgs , profila mainīšana , pedējie jautājumi un pogas orientācijai vietnē- (7.3. att)



7.3. attēls - galvena lappa.

Šeit ir iespēja rediģēt profilu un to datus, vārda un uzvārda un pasta mainīšana ir pieejama.

Arī ir iespēja samainīt bildi lietotājam. (7.4. att)



The image shows a user profile editing interface. On the left is a small preview of the profile card. The main area on the right is the editing form. At the top of the form is a large header image with a circular avatar placeholder and a blue 'Change Avatar' button. Below this is the username 'QWE'. The form contains several input fields: 'First Name' (with 'qwe'), 'Last Name' (with 'qwe'), 'Email address' (with 'qwe123@gmail.com'), 'Profile Headline' (with 'qwe'), and 'Profile Bio' (with 'qwe'). A 'Gender' section has two radio buttons, with the 'F' (Female) option selected. A small disclaimer 'We'll never share your email with anyone else.' is located below the email field.

Qwe
Qwe Qwe
qwe

Change Avatar

QWE

qwe First Name qwe Last Name

Email address
qwe123@gmail.com
We'll never share your email with anyone else.

Gender
☒ F

Profile Headline
qwe

Profile Bio
qwe

7.4. attēls- datu rediģēšana.

Ir iespēja mainīt paroli, datus par sevi. Pēc mainīšanas jānospiež pogu – **update profile.** -
(7.5. att)

We'll never share your email with anyone else.

Gender

☐ M ☒ F

Profile Headline

qwe

Profile Bio

qwe

Change Password

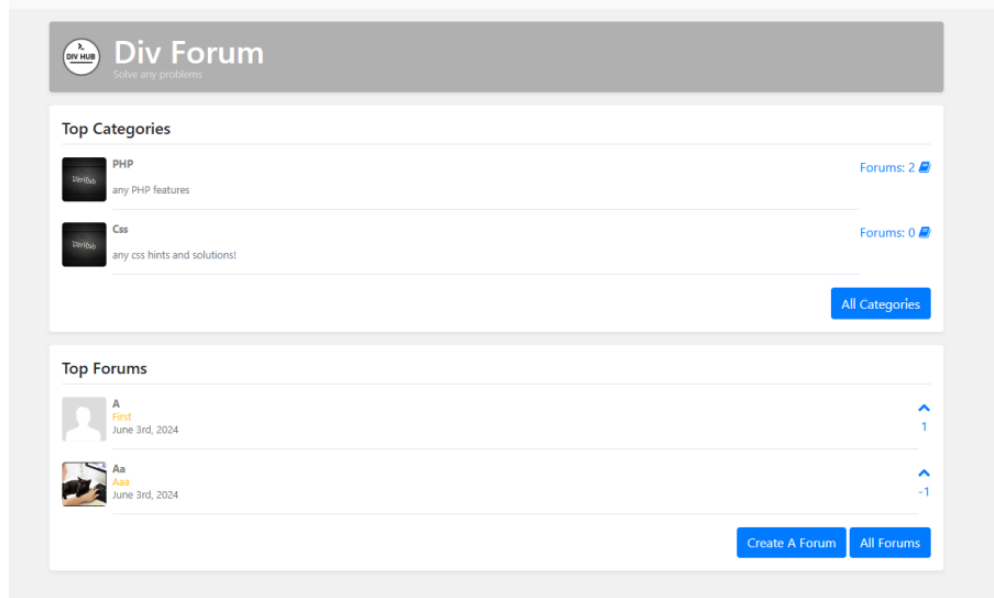
Current Password

New Password Repeat New Password

Update Profile

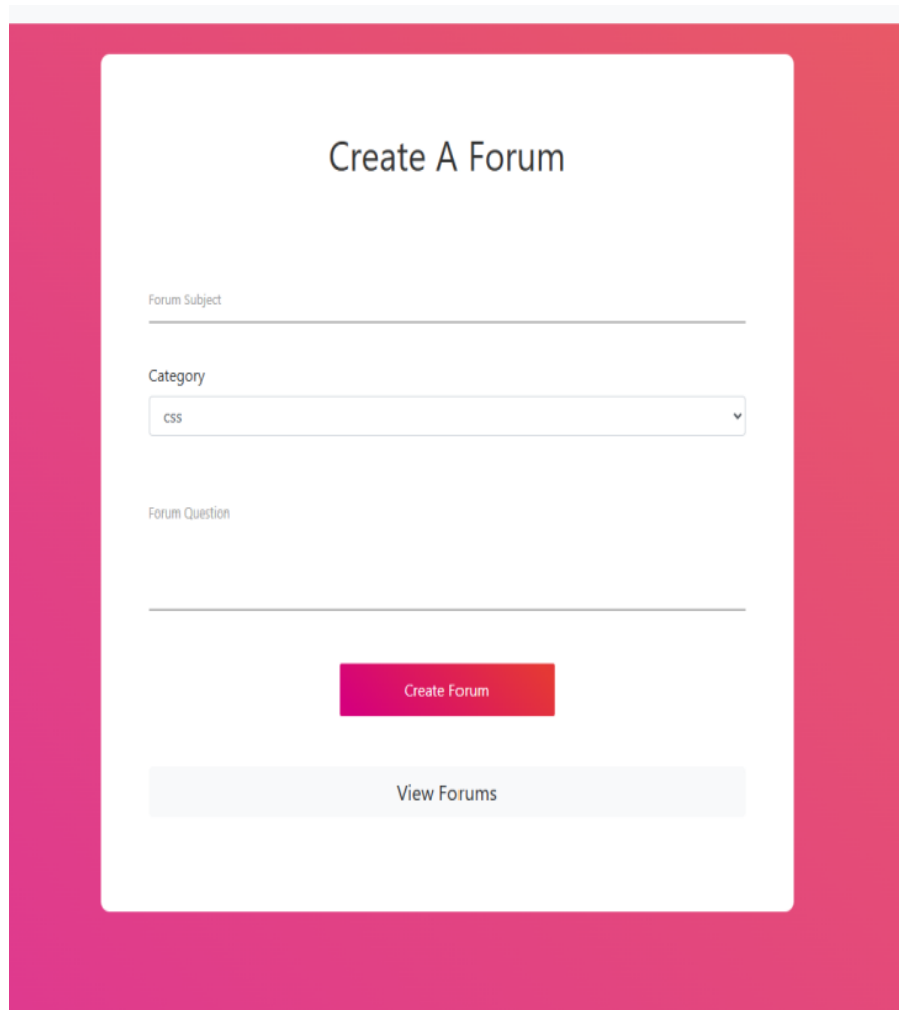
7.5. attēls - paroles maiņa.

Kategorijas un izveidotie jautājumi , jūs varat apskatīt jaunākos jautājumus un pēc kategorijas samēklēt vajadzīgo jautājumu.-(7.6. att)



7.6.attēls - Kategorijas un jautājumi.

Ir iespēja uzdot jautājumu un izvēlēties vajadzīgo kategoriju. Pirmajā laukā ir jautājuma tēma, otrajā laukā var izvēlēties kategoriju un pēdējā laukā uzrakstīt pašu jautājumu - (7.7. att)

A screenshot of a web form titled "Create A Forum". The form is centered on a white background with a thick pink border. It contains three input fields: "Forum Subject", "Category" (a dropdown menu with "CSS" selected), and "Forum Question". Below the inputs are two buttons: a red "Create Forum" button and a light gray "View Forums" button.

Create A Forum

Forum Subject

Category

CSS

Forum Question

Create Forum


View Forums

7.7. attēls - jautājumu veidošana.

Ir iespēja veidot jaunus komentarus ,dzēst tos un balsot.Zēm profila ir poga lai parskatīt lietotāja profilu.Centrā tik parādīts pats komentars. Labajā pusē ir dzēšanas poga un balsojumu pogas.-(7.8.att)

June 5th, 2024

#5




first
aa


Joined: Jun 5, 2024


Messages: 3

Likes: 0




:3





0



DIV Forum

Submit reply

7.8. attēls - komentaru veidošana.

NOBEIGUMS

Sistēmas izstrādes gaitā tika izpildīti vairāki uzdevumi un atrisinātas problēmas, un projekts šobrīd ir gandrīz pabeigts. Lietotājam draudzīgums, vienkāršs interfeiss un optimizācija pat zemas klases ierīcēm bija galvenais projektēšanas un izstrādes procesā. Tehnoloģijas strauji attīstās, un ir nepieciešami pastāvīgi uzlabojumi un jauninājumi. Darbā tika izmantotas atbilstošas tehnoloģijas un rīki datu bāzes operāciju veikšanai, izmantojot SQL Server, nodrošinot elastību turpmākajām platformas izmaiņām un uzlabojumiem.

Projekta izstrādes gaitā tika izmantots Visual Studio Code, kas piedāvā efektīvu programmatūras izstrādes vidi. Lietotāja saskarne ir izstrādāta, izmantojot HTML5, CSS3 un JavaScript, lai nodrošinātu optimālu lietotāja pieredzi tīmeklī. Šīs valodas un tehnoloģijas ļauj izveidot interaktīvas un bagātīgas lietotāja saskarnes. Katrs no šiem rīkiem un tehnoloģijām tika pielietots konkrētiem uzdevumiem vai moduļiem, nodrošinot daudzpusīgu pieeju projektam.

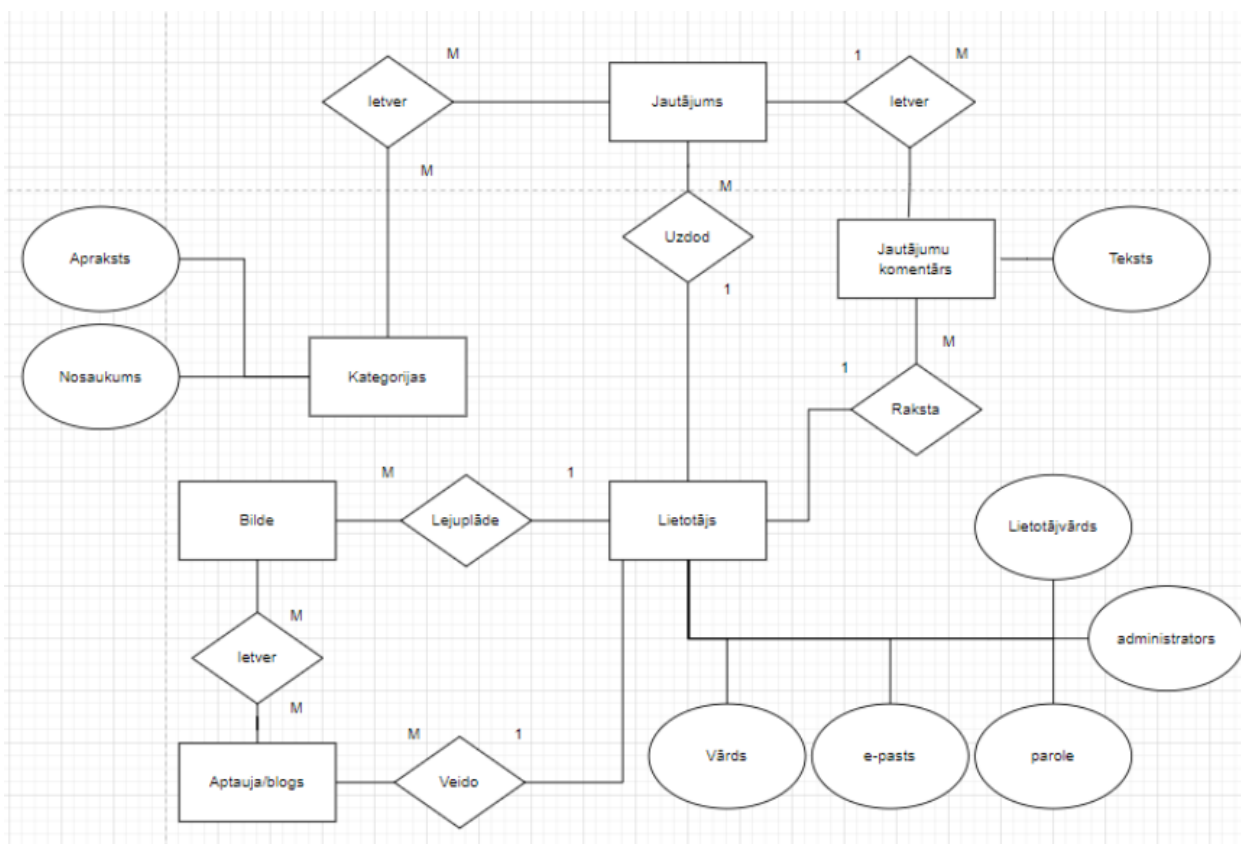
Sākotnēji bija problēmas ar atsevišķu funkciju izstrādi un ieviešanu. bet jo vairāk iedziļinājos projektā, jo labāk tas sanāca!

INFORMĀCIJAS AVOTI

- JavaScript (JS) dokumentācija -
<https://developer.mozilla.org/en-US/docs/Web/JavaScript> - (Resurss apskatīts 20.04.2024)
- PHP dokumentācija - <https://www.php.net/manual/en/> - (Resurss apskatīts 20.05.2024)
- CSS dokumentācija - <https://developer.mozilla.org/en-US/docs/Web/CSS> - (Resurss apskatīts 15.5.2024)
- MySQL (SQL) dokumentācija - <https://dev.mysql.com/doc/> - (Resurss apskatīts 01.05.2024)
- Github - https://github.com/rvt-prog-kval-24/programm-taju_soci-lais_tikls- (Resurss apskatīts 1.06.2024)
- Bootstrap - <https://getbootstrap.com/docs/5.0/getting-started/introduction/> - (Resurss apskatīts 22.05.2024)

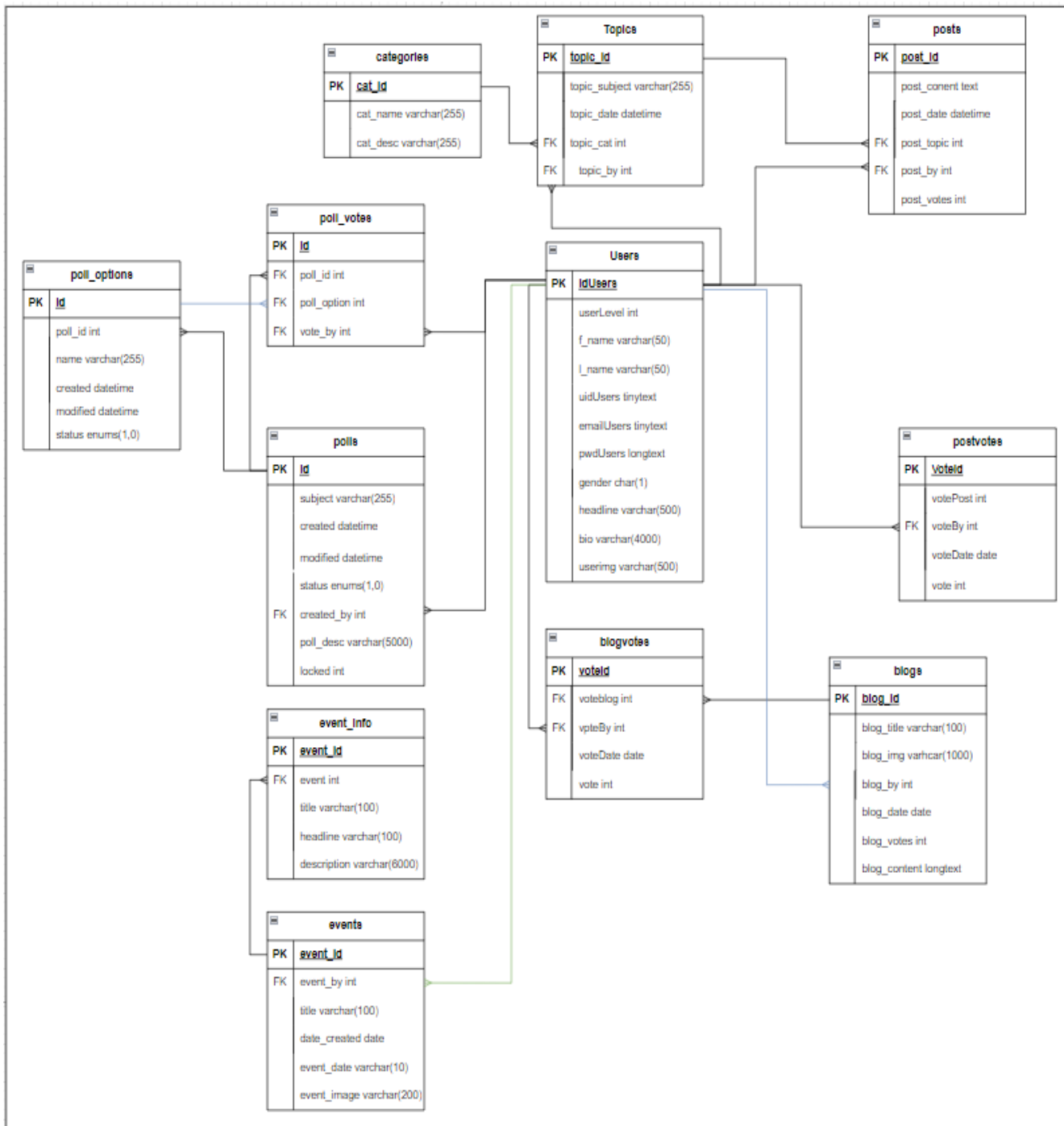
PIELIKUMI

1. pielikums



ER-modelis

2. pielikums



Tabulu relāciju shēma

Reģistrācijas

kods:

```
<?php
    if (isset($_POST['signup-submit']))
    {
        require 'dbh.inc.php';
        $userName = $_POST['uid'];
        $email = $_POST['mail'];
        $password = $_POST['pwd'];
        $passwordRepeat = $_POST['pwd-repeat'];
        $gender = $_POST['gender'];
        $headline = $_POST['headline'];
        $bio = $_POST['bio'];
        $f_name = $_POST['f-name'];
        $l_name = $_POST['l-name'];
        if (empty($userName) || empty($email) || empty($password) || empty($passwordRepeat))
        {

            header("Location: ../signup.php?error=emptyfields&uid=".$userName."&mail=".$email);
            exit();
        }
        else if (!filter_var($email, FILTER_VALIDATE_EMAIL) && !preg_match("/^[a-zA-Z0-9]*$/", $userName))
        {
            header("Location: ../signup.php?error=invalidmailuid");
            exit();
        }
        else if (!filter_var($email, FILTER_VALIDATE_EMAIL))
        {
            header("Location: ../signup.php?error=invalidmail&uid=".$userName);
            exit();
        }
    }
}
```



```

    }
    else if (!preg_match("/^[a-zA-Z0-9]*$/", $userName))
    {
        header("Location: ../signup.php?error=invaliduid&mail=".$email);
        exit();
    }
    else if ($password !== $passwordRepeat)
    {

header("Location: ../signup.php?error=passwordcheck&uid=".$userName."&mail=".$email);
        exit();
    }
    else
    {
        // checking if a user already exists with the given username
        $sql = "select uidUsers from users where uidUsers=?;";
        $stmt = mysqli_stmt_init($conn);
        if (!mysqli_stmt_prepare($stmt, $sql))
        {
            header("Location: ../signup.php?error=sqlerror");
            exit();
        }
        else
        {
            mysqli_stmt_bind_param($stmt, "s", $userName);
            mysqli_stmt_execute($stmt);
            mysqli_stmt_store_result($stmt);

            $resultCheck = mysqli_stmt_num_rows($stmt);

            if ($resultCheck > 0)

```

```

{
    header("Location: ../signup.php?error=usertaken&mail=".$email);
    exit();
}
else
{
    $FileNameNew = 'default.png';
    require 'upload.inc.php';

    $sql = "insert into users(uidUsers, emailUsers, f_name, l_name, pwdUsers,
gender, "
        . "headline, bio, userImg) "
        . "values (?,?,?,?,?,?,?,?,?)";
    $stmt = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmt, $sql))
    {
        header("Location: ../signup.php?error=sqlerror");
        exit();
    }
    else
    {
        $hashedPwd = password_hash($password, PASSWORD_DEFAULT);

        mysqli_stmt_bind_param($stmt, "ssssssss", $userName, $email, $f_name,
$l_name,
            $hashedPwd, $gender,
            $headline, $bio, $FileNameNew);
        mysqli_stmt_execute($stmt);
        mysqli_stmt_store_result($stmt);

        header("Location: ../signup.php?signup=success");
    }
}

```

```

        exit();
    }
}
}

mysqli_stmt_close($stmt);
mysqli_close($conn);

}

else
{
    header("Location: ../signup.php");
    exit();
}3

```

<?php

```
session_start();
```

```
define('TITLE',"Create Category | DiV");
```

```
if(!isset($_SESSION['userId']))
{
    header("Location: login.php");
    exit();
}
```

```
include 'includes/HTML-head.php';
```

?>

<link rel="stylesheet" type="text/css" href="css/comp-creation.css">

</head>

<body>

```
<?php include 'includes/navbar.php'; ?>
```

<div class="bg-contact2" style="background-image: url('img/black-bg.jpg');">

<div class="container-contact2">

<div class="wrap-contact2">

```
<form class="contact2-form" method="post" action="includes/create-  
category.inc.php">
```


Create Category

<?php

```
if(isset($_GET['error']))
```

$$\{$$

```
if($_GET['error'] == 'emptyfields')
```

$$\{$$

```
echo '<h5 class="text-danger">*Fill In All The Fields</h5>';
```

$$\}$$

```
else if ($_GET['error'] == 'catnametaken')
```

$$\{$$

```
echo '<h5 class="text-danger">*A category with the given
```

name already exists</h5>';

$$\}$$

```
else if ($ GET['error'] == 'sqlerror')
```

```

        {
            echo '<h5 class="text-danger">*Website Error: Contact admin
to have the issue fixed</h5>';
        }
    }
    else if (isset($_GET['catcreation']) == 'success')
    {
        echo '<h5 class="text-success">*Category successfully
created</h5>';
    }
?>
</span>

<div class="wrap-input2 validate-input" data-validate="Name is required">
    <input class="input2" type="text" name="cat_name">
    <span class="focus-input2" data-placeholder="Category Name"></span>
</div>

<div class="wrap-input2 validate-input" data-validate = "Description is required">
    <textarea class="input2" name="cat_description"></textarea>
    <span class="focus-input2" data-placeholder="Category Description"></span>
</div>

<div class="container-contact2-form-btn">
    <div class="wrap-contact2-form-btn">
        <div class="contact2-form-bgbtn"></div>
        <button class="contact2-form-btn" type="submit"
name="create-cat">
            Add Category
        </button>
    </div>
</div>

    <div class="text-center">
        <br><br><a class="btn btn-light btn-lg btn-block"
href="categories.php">
            View Categories</a>
    </div>

</form>
</div>
</div>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/creation-main.js"></script>
</body>

```

</html>

5.pielikums

Blogu veidošana

```
<?php
    session_start();
    define('TITLE','Create Blog | :3');

    if(!isset($_SESSION['userId']))
    {
        header("Location: login.php");
        exit();
    }

    include 'includes/HTML-head.php';
?>
    <link rel="stylesheet" type="text/css" href="css/comp-creation.css">
</head>

<body>
    <?php include 'includes/navbar.php'; ?>
    <div class="bg-contact2" style="background-image: url('img/black-bg.jpg');">
        <div class="container-contact2">

            <div class="wrap-contact2">
                <form class="contact2-form" action="includes/create-blog.inc.php"
method='post'
                enctype="multipart/form-data">

                    <label class="btn btn-primary position-absolute mt-2 ml-2">
                        Change Cover Image <input type="file" id="imgInp" name='dp'
hidden>

                    </label>
                    

                    <br><br><br>
                    <span class="contact2-form-title">
                        Create a Blog
                    </span>

                    <span class="text-center">
                        <?php
                            if(isset($_GET['error']))
                            {
```

```

        if($_GET['error'] == 'emptyfields')
        {
            echo '<h5 class="text-danger">*Fill In All The Fields</h5>';
        }
        else if ($_GET['error'] == 'catnametaken')
        {
            echo '<h5 class="text-danger">*A category with the given
name already exists</h5>';
        }
        else if ($_GET['error'] == 'sqlerror')
        {
            echo '<h5 class="text-danger">*Website Error: Contact admin
to have the issue fixed</h5>';
        }
    }
    else if (isset($_GET['catcreation']) == 'success')
    {
        echo '<h5 class="text-success">*Category successfully
created</h5>';
    }
    ?>
</span>

```

```

<div class="wrap-input2 validate-input" data-validate="Name is required">
    <input class="input2" type="text" id="title" name="btitle">
    <span class="focus-input2" data-placeholder="Blog Title"></span>
</div>

<div class="wrap-input2 validate-input" data-validate = "Description is required">
    <textarea class="input2" id="content"
name="bcontent"rows="20"></textarea>
    <span class="focus-input2" data-placeholder="Blog Content"></span>
</div>

<div class="container-contact2-form-btn">
    <div class="wrap-contact2-form-btn">
        <div class="contact2-form-bgbtn"></div>
        <button class="contact2-form-btn" type="submit"
name="create-blog-submit">
            Create Blog
        </button>
    </div>
</div>

<div class="text-center">
    <br><br><a class="btn btn-light btn-lg btn-block" href="blogs.php">
        View All Blogs</a>

```

```

        </div>
    </form>
</div>
</div>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/creation-main.js"></script>

<script>
    var dp = 'img/blog-cover.png';

    $('#blah').attr('src', dp);

    function readURL(input) {

        if (input.files && input.files[0]) {
            var reader = new FileReader();

            reader.onload = function(e) {
                $('#blah').attr('src', e.target.result);
            }

            reader.readAsDataURL(input.files[0]);
        }
    }

    $("#imgInp").change(function() {
        readURL(this);
    });

</script>

</body>
</html>

```

8.Pielikums
Aptauju veidošana

```

<?php

    session_start();

    define('TITLE','Create Poll | Div');

```



```

    if(!isset($_SESSION['userId']))
    {
        header("Location: login.php");
        exit();
    }

    include 'includes/HTML-head.php';
?>
    <link rel="stylesheet" type="text/css" href="css/comp-creation.css">
</head>
<body>
    <?php include 'includes/navbar.php'; ?>

    <div class="bg-contact2" style="background-image: url('img/black-bg.jpg');">
        <div class="container-contact2">
            <div class="wrap-contact2">
                <form class="contact2-form" method="post" action="includes/create-poll.inc.php">
                    <span class="contact2-form-title">
                        Create Poll
                    </span>

                    <span class="text-center">
                        <?php
                            if(isset($_GET['error']))
                            {
                                if($_GET['error'] == 'emptyfields')
                                {
                                    echo '<h5 class="text-danger">*Fill In All The Fields</h5>';
                                }
                                else if ($_GET['error'] == 'titletaken')
                                {
                                    echo '<h5 class="text-danger">*There is already a poll with
this title</h5>';
                                }
                                else if ($_GET['error'] == 'sqlerror')
                                {
                                    echo '<h5 class="text-danger">*Website Error: Contact admin
to have the issue fixed</h5>';
                                }
                            }
                        else if (isset($_GET['creation']) == 'success')
                        {
                            echo '<h5 class="text-success">*Poll successfully created</h5>';
                        }
                    </span>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="wrap-input2 validate-input" data-
validate="Name is required">
        <input class="input2" type="text" name="title">
        <span class="focus-input2" data-placeholder="Poll Title"></span>
    </div>

    <div class="checkbox-animated">
        <input id="checkbox_animated_1" type="checkbox" name="is-
locked" value="is-locked">
        <label for="checkbox_animated_1">
            <span class="check"></span>
            <span class="box text-muted"></span>
            Make the poll Locked
        </label>
    </div><br>

    <div class="wrap-input2 validate-input" data-validate = "Description is required">
        <textarea class="input2" name="desc"></textarea>
        <span class="focus-input2" data-placeholder="Poll Description"></span>
    </div>

    <div class="col-sm-4">
        <label for="option">Poll Options</label>
        <div class="input_fields_wrap">
            <button class="add_field_button btn btn-light">Add More
Fields</button>

            <div class="wrap-input2 ">
                <input class='input2' type="text" name="option[]"
id="option" placeholder="poll option">
            </div>
            <div class="wrap-input2 ">
                <input class='input2' type="text" name="option[]"
id="option" placeholder="poll option">
            <div class="container-contact2-form-btn">
                <div class="wrap-contact2-form-btn">
                    <div class="contact2-form-bgbtn"></div>
                    <button class="contact2-form-btn" type="submit"
name="poll-submit">
                        Add Poll
                    </button>
                </div>
            </div>
        </div>

        <div class="text-center">
            <br><br><a class="btn btn-light btn-lg btn-block" href="polls.php">
                View Polls</a>
        </div>

```

```

        </div>

    </form>

</div>
</div>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/creation-main.js"></script>
<script>
$(document).ready(function() {
    var max_fields    = 6; //maximum input boxes allowed
    var wrapper      = $(".input_fields_wrap"); //Fields wrapper
    var add_button    = $(".add_field_button"); //Add button ID

    var x = 1; //initlal text box count
    $(add_button).click(function(e){ //on add input button click
        e.preventDefault();
        if(x < max_fields){ //max input box allowed
            x++; //text box increment
            $(wrapper).append('<div class="wrap-input2">\n\
                <input type="text" name="option[]" placeholder="poll option"
id="option"\
                ><a href="#" class="remove_field"><i class="fa fa-trash"
aria-hidden="true"></i></a>\
                </div>'); //add input box
            $(wrapper).on("click", ".remove_field", function(e){ //user click on remove text
                e.preventDefault(); $(this).parent('div').remove(); x--;
            });
        }
    });
</script>

</body>
</html>

```