**RĪGAS VALSTS TEHNIKUMS**

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

**KVALIFIKĀCIJAS DARBS**

**“Elektropreču internetveikala datu uzskaites sistēma”**

Paskaidrojošais raksts 37. lpp.

Audzēknis: Kristaps Ešmits

Prakses vadītājs: Ilona Demčenko

Nodaļas vadītājs: Normunds Barbāns

**Rīga 2023**

**ANOTĀCIJA**

Kvalifikācijas darba uzdevums ir izveidot “Elektropreču internetveikala datu uzskaites sistēmu”, ar kuru jebkurš cilvēks spēs pilnvērtīgi darboties neatkarīgi vai ir apmācīts, vai nav apmācīts darboties ar konkrētos sistēmu. Sistēma tiek veidota pēc konkrēta uzņēmuma pasūtījuma. Backend daļas izstrādei tika izvēlēts PHP, Laravel, Axios. Servera daļai tiek izmantots XAMPP, Apache. Lietotāja saskarnei tika izvēlēts Vue.js , Vite.js, Bootstrap. Priekš datu bāzes tiek izmantots MySQL ar phpmyadmin.

Kvalifikācijas darba paskaidrojošais raksts satur – ievadu, uzdevuma nostādni, prasību specifikāciju, uzdevuma risināšanas līdzekļu izvēles pamatojumu, programmatūras produkta modelēšanas un projektēšanas aprakstu, datu struktūru aprakstu, lietotāja ceļvedi, nobeigumu un pielikumus. Ievadā ir aprakstīts sistēmas nepieciešamība, aktualitāte un lietderīgums. Uzdevuma nostādnē ir aprakstīts kvalifikācijas darba galvenais izveidošanas mērķis un tā uzdevumi. Prasību specifikācijā ir aprakstītas sistēmas funkcionālās un nefunkcionālās prasības, kā arī aplūkojama sistēmas izejas un ieejas informācija – dati, ko lietotājs ievada un dati, kuri tiek izvadīti lietotājam. Uzdevuma risināšanas līdzekļu izvēles pamatojumā ir aprakstīts, kādas programmēšanas valodas, teksta redaktori, datu bāzes, rīki tika izmantoti sistēmas izstrādē. Programmatūras produkta modelēšanas un projektēšanas aprakstā ir apskatāma datu plūsmu diagramma, ER diagramma, datu bāzes uzbūve, kā arī aprakstīta sistēmas arhitektūra. Datu struktūru apraksts satur visu tabulu struktūru un to aprakstu, kā arī tabulu relāciju shēmu. Lietotāja ceļvedis detalizēti attēlo sistēmas informācijas vizuālo izkārtojumu un paskaidro kā pareizi lietot sistēmu. Nobeigumā ir aprakstīts, kas beigās tika izveidots.

Kvalifikācijas darba apjoms ir ..lpp, kurā ietilps … attēli, … tabulas un … pielikumi.

**aNNOTATION**

The task of the qualification work is to create a "data accounting system of an electrical goods online store", with which any person will be able to fully function, regardless of whether he is trained or not trained to operate the specific system. The system is created according to the order of a specific company. PHP, Laravel, Axios were chosen for the development of the backend part. XAMPP, Apache is used for the server part. Vue.js, Vite.js, Bootstrap were chosen for the user interface. MySQL with phpmyadmin is used for the database.

The explanatory article of the qualification work contains – introduction, statement of the task, specification of requirements, justification for choosing means of solving the task, description of software product modelling and design, description of data structures, users guide, conclusion and appendices. The introduction describes the need, relevance and usefulness of the system. The assignment statement describes the main purpose of creating the qualification work and its tasks. The requirements specification describes the functional and non-functional requirements of the system, as well as the output and input information of the system - data that is entered by the user and data that is output to the user. The justification for choosing the tools for solving the task describes what programming languages, text editors, databases, and tools were used in the development of the system. The software product modelling and design description includes a data flow diagram, an ER diagram, a database structure, as well as a described system architecture. The description of data structures contains the structure of all tables and their description, as well as the relational schema of the tables. The user guide shows the visual layout of the system information in detail and explains how to use the system correctly. The conclusion describes what was created in the end.

The scope of the qualification work is ..pages, which includes ... images, ... tables and ... appendices.

**Saturs**

[**ievads 5**](#_Toc167570712)

[**1. UZDEVUMA NOSTĀDNE 6**](#_Toc167570713)

[**2. PRASĪBU SPECIFIKĀCIJA 7**](#_Toc167570714)

[2.1. Ieejas un izejas informācijas apraksts 7](#_Toc167570715)

[2.1.1. Ieejas informācijas apraksts 7](#_Toc167570716)

[2.1.2. Izejas informācijas apraksts 9](#_Toc167570717)

[2.2. Funkcionālās prasības 10](#_Toc167570718)

[2.3. Nefunkcionālās prasības 13](#_Toc167570719)

[**3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS 15**](#_Toc167570720)

[**4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA 16**](#_Toc167570721)

[4.1. Sistēmas struktūras modelis 16](#_Toc167570722)

[4.1.1. Sistēmas arhitektūra 16](#_Toc167570723)

[4.1.2. Sistēmas ER-modelis 17](#_Toc167570724)

[4.2. Funkcionālais sistēmas modelis 19](#_Toc167570725)

[4.2.1. Datu plūsmu modelis 19](#_Toc167570726)

[**5. DATU STRUKTŪRU APRAKSTS 23**](#_Toc167570727)

[**6. LIETOTĀJA CEĻVEDIS 28**](#_Toc167570728)

[6.1. Sistēmas prasības aparatūrai un programmatūrai 28](#_Toc167570729)

[6.2. Sistēmas instalācija un palaišana 28](#_Toc167570730)

[6.3. Programmas apraksts 28](#_Toc167570731)

[6.4. Testa piemērs 28](#_Toc167570732)

[**NOBEIGUMS 29**](#_Toc167570733)

[**INFORMĀCIJAS AVOTI 30**](#_Toc167570734)

[**PIELIKUMI 31**](#_Toc167570735)

[1.pielikums. Pārskatu piemēri 31](#_Toc167570736)

[2.pielikums. Ekrānattēlu piemēri 31](#_Toc167570737)

[3.pielikums. Programmas pirmteksts 31](#_Toc167570738)

# ievads

Mūsdienās ir ļoti svarīgi izmantot modernas datu uzskaites sistēmas, jo informācijas un digitālās tehnoloģijas attīstās straujāk nekā jebkad, tāpēc ir nepieciešamas sistēmas, kas atbilst mūsdienu standartiem, ir viegli pielāgojamas un ātrdarbīgas*.* Šī kvalifikācijas darba mērķis ir izveidot “Elektropreču internetveikala datu uzskaites sistēmu”, ar kuru jebkurš cilvēks spēs pilnvērtīgi darboties neatkarīgi vai ir apmācīts, vai nav apmācīts darboties ar konkrētos sistēmu. Sistēma tiek veidota pēc konkrēta uzņēmuma pasūtījuma, bet to ir iespējams arī pielāgot citiem uzņēmumiem.

Pirms datus uzskaites sistēmas izstrādes tiek veikta tirgus izpēte, lai saprastu, kādi produkti jau atrodas pašreizējā tirgū, to trūkumi un priekšrocības, kā arī kāpēc pasūtītājs vēlas pilnīgi jaunu datu uzskaites sistēmu, nevis kādu no tirgū jau esošajām datu uzskaites sistēmām. Esošie analogi klientu neapmierini, jo tie ir ļoti dārgi, it īpaši biznesam paplašinoties. Lai sasniegtu pasūtītāja velmes būtu nepieciešams implementēt divas dažādas jau esošas sistēmas. Viena, kas atbildētu par noliktavas pārvaldību, otra kas atbildētu par internetveikala darbību, tādējādi ļoti sadārdzinot izmaksas. Problēma izmantojot pašreizējos analogus ir tāda, ka ir nepieciešams apmācības periods, kā arī tie neatbilst līdz galam pasūtītāja velmēm. Viens no analogu piemēriem ir “SAP Inventory Management”. Šī sistēma atbilst daudziem no pasūtītajā punktiem - automatizēta noliktavas darbība, analītiskie rīki, bet tai ir augstas izmaksas, sarežģīta implementācija, sarežģītā pielāgošana un ilgs apmācību periods, kā arī tā atbild tikai par noliktavas daļu, tas nozīmē, ka ir nepieciešams implementēt vēl vienu sistēmu, kā ” Magento”, kura atbildēs par internetveikala darbību.

Pašreizējai datu uzskaites sitamai pasūtītājs izmanto Excel, tādējādi bieži vien uzņēmums saskaras ar ierobežojumiem, kas ietekmē uzņēmumu efektivitāti un attīstību. Uzņēmumam augot datu apjoms palielinās, tādējādi ir grūti apkopot visus datus, kā arī Excel pie liela datu apjoma kļūst lēns, tādējādi datu apstrādes laiks saildzinās un pat rodas neprecizitātes.

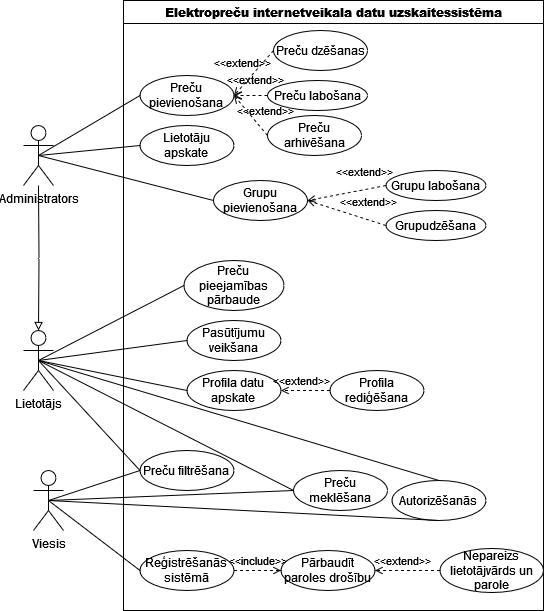
Izmantojot “Elektropreču internetveikala datu uzskaites sistēmu” pasūtītājs atvieglo dzīvi darbiniekiem, jo sistēmas izmantošanai nevajag apmācību, tā nodrošina visu nepieciešamos līdzekļus, kā arī ietaupa līdzekļus, ja salīdzina ar tirgū esošajiem analogiem.

# UZDEVUMA NOSTĀDNE

Kvalifikācijas darba uzdevums ir izveidot Elektropreču internetveikala datu uzskaites sistēmu. Sistēmā nepieciešams realizēt iespēju sistēmas administratoram veikt uzskaiti un darbības ar precēm, kuras pieder veikalam, kā arī radīt iespēju potenciālajiem pircējiem iespēju aplūkot un iegādāties kādu no precēm.

Elektropreču internetveikala datu uzskaites sistēma ir aktuāla konkrētam pasūtītājam, lai uzlabotu sava veikala, noliktavas darbību, kā arī veikala apmeklētājiem, tādējādi dodot tiem iespēju iepirkties un uzzināt informāciju gan par precēm, gan par veikalu neizejot no mājas. Pašreizējie tirgū pieejamie interneta veikali neefektīvi prezentē administratīvos datus un klientu informāciju. Sistēma radītu intuitīvāku sadarbību starp klientiem un administratoriem caur skaidrāku datu vizualizāciju. Mērķauditorija ir ne tikai konkrētais pasūtītājs un tā klientu loks, bet arī citi uzņēmumi, kuriem ir iespējams piemērot šādu uzskaites sistēmu.

Elektropreču internetveikala datu uzskaites sistēmai ir jāizpilda sekojošas funkcionalitātes (skat. 1.1. att.):

* preču pievienošana, labošana, dzēšana, arhivēšana;
* pasūtījumu veikšana;
* preču meklēšana;
* preču filtrēšana pēc parametriem;
* preču pieejamības pārbaude;
* lietotāja reģistrēšana, autorizēšana, datu labošana, profila apskate.
  1. att. Lietojumgadījuma diagramma

# PRASĪBU SPECIFIKĀCIJA

## Ieejas un izejas informācijas apraksts

### **Ieejas informācijas apraksts**

Sistēmā tiks nodrošināta šādas ieejas informācijas apstrāde:

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

* Vārds – lietotajā vārds – burtu teksts ar izmēru līdz 30 rakstzīmēm, piemēram - Jānis;
* Uzvārds – lietotāja uzvārds – burtu teksts ar izmēru līdz 30 rakstzīmēm, piemēram - Bērziņš;
* Parole – parole, ko lietotājs izmantos, lai ienāktu sistēmā – burtu teksts ar izmēru līdz 255 rakstzīmēm, piemēram - JanisB1987;
* Telefona numurs – lietotāja telefona numurs – cipars ar izmēru līdz 15 rakstzīmēm, piemēram 25677359;
* E-pasts – lietotāja e-pasts – burtu teksts ar izmēru līdz 255 rakstzīmēm, piemēram - janisberzins@inbox.lv;
* Loma – lietotajā loma, viesis, administrators, vai reģistrēts lietotājs automātiski reģistrēts lietotājs – cipars ar izmēru līdz 1 rakstzīmei, piemēram - 1;
* Izveidots – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13;
* Atjaunots – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13.

\* Visu informāciju par **Lietotājiem** ievadīs lietotājs no klaviatūras izņemot laukus izveidots un atjaunināts, tos iegūs no API.

2. Informācija par **Kategorijām** sastāvēs no šādiem datiem.

* Nosaukums – Kategorijas nosaukums – burtu teksts ar izmēru līdz 50 rakstzīmēm, piemēram - Klaviatūras;
* Bilde – preces bilde – burtu teksts ar izmēru līdz 255 rakstzīmēm, piemēram - EVk7LOgI2G.png
* Izveidots – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13;
* Atjaunots – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13.

\* Visu informāciju par **Kategorijām** ievadīs lietotājs no klaviatūras.

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

* Nosaukums – preces nosaukums – burtu teksts ar izmēru līdz 255 rakstzīmēm, piemēram- Klaviatūra mikro pele;
* Apraksts – preces apraksts – burtu teksts ar izmēru līdz 5000 rakstzīmēm, piemēram - Jaunākā tipa klaviatūra ar viesām nepieciešamajām funkcijām;
* Cena – preces cena – daļskaitlis ar precizitāti līdz 2 cipariem aiz komata, piemēram - 90.99;
* Bilde – preces bilde – burtu teksts ar izmēru līdz 255 rakstzīmēm, piemēram - EVk7LOgI2G.png;
* Izveidots – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13;
* Atjaunots – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13.

\* Visu informāciju par **Precēm** ievadīs lietotājs no klaviatūras.

4. Informācija par **Grozu** sastāvēs no šādiem datiem.

* Status – grozs ir aktīvs vai neaktīvs – patiess/nepatiess, piemēram - Patiess;
* Kopējais preču daudzums – cik daudz preces ir grozā – cipars ar izmēru līdz 11 rakstzīmēm, piemēram - 5;
* Cena – Kopējā cena, cik par precēm ir jāmaksā – cipars ar izmēru līdz 11 rakstzīmēm, piemēram - 10.50.

\* Visu informāciju par **Grozu** iegūst no API.

6.. Informācija par **Zīmoliem** sastāvēs no šādiem datiem.

* Nosaukums – zīmola nosaukums – burtu teksts ar izmēru līdz 255 rakstzīmēm, piemēram- Klaviatūra mikro pele;
* Bilde – zīmola logo – burtu teksts ar izmēru līdz 255 rakstzīmēm, piemēram - EVk7LOgI2G.png;
* Izveidots – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13;
* Atjaunots – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13.

\* Visu informāciju par **Zīmoliem** ievadīs lietotājs no klaviatūras.

5. Informācija par **Pasūtījums** sastāvēs no šādiem datiem.

* Kopējā cena - Kopējā cena, cik par precēm ir jāmaksā – cipars ar izmēru līdz 11 rakstzīmēm, piemēram - 10.50.
* Status – Pasūtījuma status vai pasūtījums ir izpildīts vai nē – burtu teksts ar izmēru līdz 255 rakstzīmēm, piemēram closed vai active.
* Pasūtījuma datums – automātiska aile ar datumu un laiku, piemēram - 2023-11-02 14:29:13.

\* Visu informāciju par **Grozu** iegūst no API.

### **Izejas informācijas apraksts**

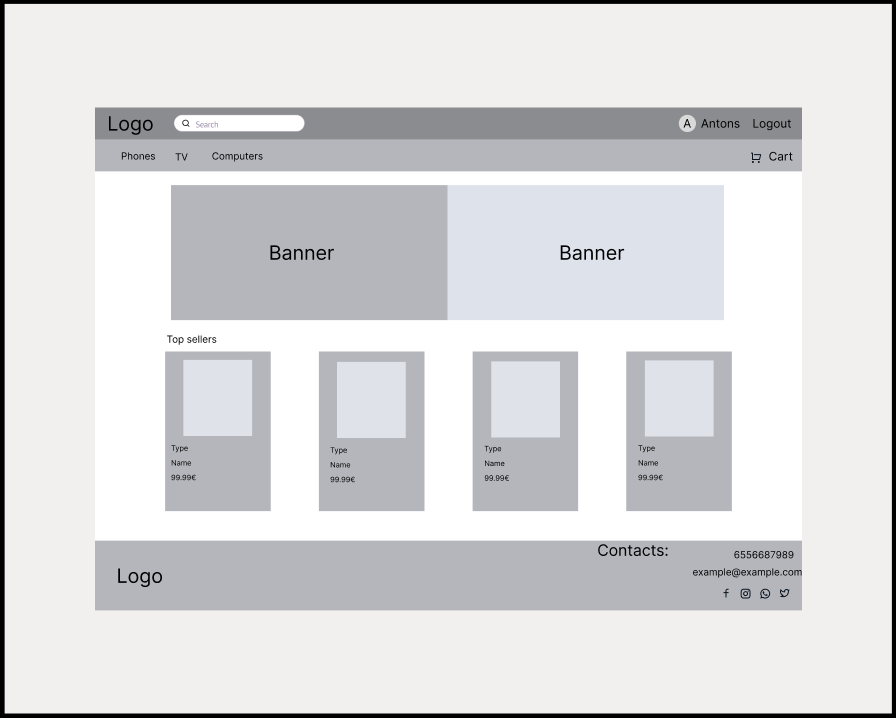
1. **Kvīts izvade PDF formātā** pēc pasūtījuma veikšanas. PDF failā tiks atspoguļota informācija par veikto iterāciju. Augšējā kreisajā stūrī būs firmas logo, labajā pusē būs norādīts pirkuma datums, faila vidū būs tabula ar detalizētu pirkuma informāciju (preces nosaukums, cena, cik samaksāts).
2. **E-pasta paziņojums** lietotājiem ar apstiprinājumu, ka pirkums ir veiksmīgs, e-pasta paziņojumā pielikumā būs kvīts PDF formātā.
3. **Statistikas paziņojums.** Statistikas paziņojums būs pieejams administratoram, lai uzzinātu informāciju par sistēmu (Cik ir reģistrēti lietotāji, cik preces sistēmā u.t.t).

## Funkcionālās prasības

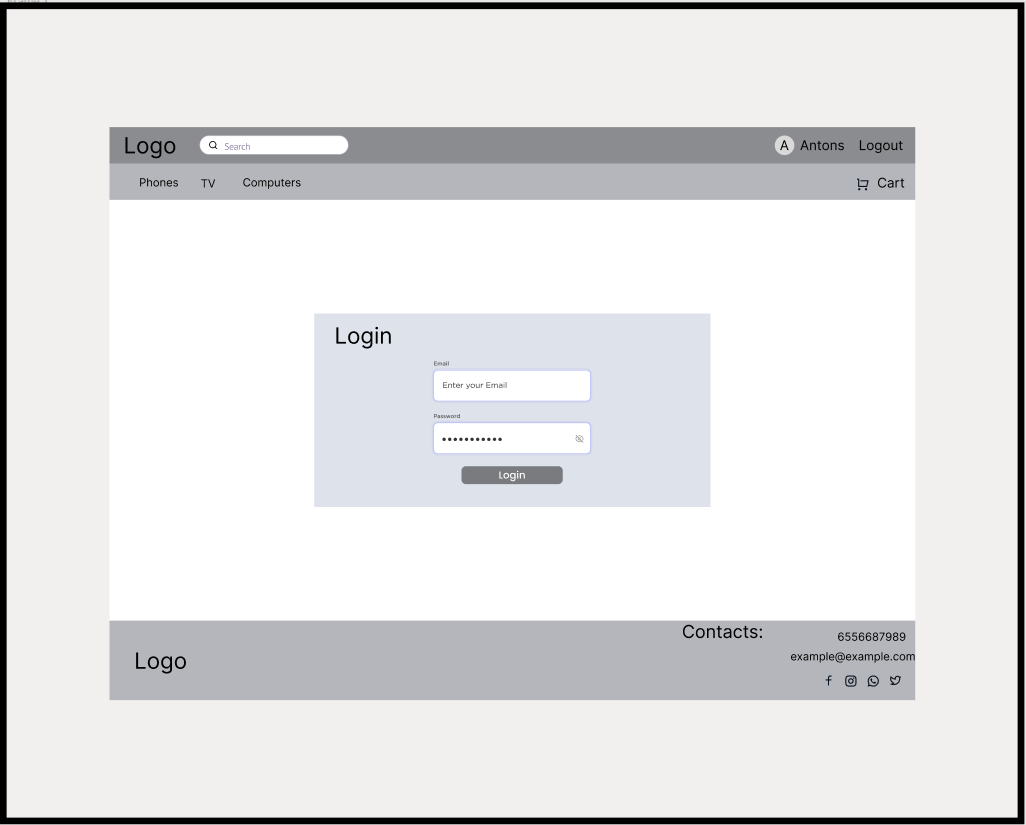
1. Jānodrošina iespēja reģistrēt jaunu lietotāju.
   1. Jāparedz ieejas informācijas par lietotāju (skat. 2.1.1. Ieejas informācijas apraksts) ievadīšana un pārbaude uz formāta pareizību.
   2. Ja kāds no obligātiem laukiem nav ievadīts, tad izvadīt par to kļūdas paziņojumu.
   3. Salīdzināt ievadīto e-pastu, telefona numuru ar sistēmā jau eksistējošo lietotāju vārdiem un izvadīt paziņojumu, ja tie sakrīt.
   4. Pārbaudīt paroli uz pietiekošu drošības pakāpi. Ja tas neatbilst, izvadīt paziņojumu.
2. Jānodrošina lietotāja autorizācija.
   1. Ja lietotāja statuss ir aktīvs sistēmai ir jānodrošina autorizācija, pieslēdzoties ar e-pastu un paroli.
   2. Ja statuss ir neaktīvs, tad sistēmai ir jāieslēdz autorizācijas lapu.
   3. Ja kāds no laukiem nav ievadīts, izvadīt par to paziņojumu.
3. Jānodrošina iespēja meklēt preces.
   1. Jāparedz iespēja meklēt preci pēc frāzes vai vārda.
4. Jānodrošina iespēja filtrēt preces.
   1. Jāparedz iespēja filtrēt preces pēc dažādām kategorijām, cenas, ražotāja, kategorijas.
5. Jānodrošina administratoram iespēju pievienot preces.
   1. Jāparedz ieejas informācijas par preci (skat. 2.1.3. Ieejas informācijas apraksts) ievadīšana un pārbaude uz formāta pareizību.
   2. Ja kāds no obligātiem laukiem nav ievadīts, tad izvadīt par to kļūdas paziņojumu.
   3. Jāparedz iespēju izvadīt paziņojumu, kad kategorija ir veiksmīgi pievienota.
6. Jānodrošina administratoram iespēju rediģēt preces.
   1. Jāparedz ieejas informācijas par preci (skat. 2.1.3. Ieejas informācijas apraksts) ievadīšana un pārbaude uz formāta pareizību.
   2. Ja prece ir veiksmīgi rediģēta izvadīt paziņojumu par veiksmīgu rediģēšanu.
   3. Ja kāds no obligātiem laukiem nav ievadīts, tad izvadīt par to kļūdas paziņojumu.
   4. Ja kāds no obligātiem laukiem neatbilst validācijas prasībām, tad izvadīt par to kļūdas paziņojumu.
7. Jānodrošina administratoram iespēju dzēst preces.
   1. Pirms preces dzēšanas ir jāizvada paziņojums, vai tiešām vēlaties dzēst konkrēto ierakstu.
8. Jānodrošina iespēju administratoram pievienot kategorijas.
   1. Jāparedz ieejas informācijas par preci (skat. 2.1.2. Ieejas informācijas apraksts) ievadīšana un pārbaude uz formāta pareizību.
   2. Ja kāds no obligātiem laukiem nav ievadīts, tad izvadīt par to kļūdas paziņojumu.
   3. Jāparedz iespēju izvadīt paziņojumu, kad kategorija ir veiksmīgi pievienota.
9. Jānodrošina iespēju administratoram rediģēt kategorijas.
   1. Jāparedz ieejas informācijas par kategoriju (skat. 2.1.2. Ieejas informācijas apraksts) ievadīšana un pārbaude uz formāta pareizību.
   2. Ja kategorija ir veiksmīgi rediģēta izvadīt paziņojumu par veiksmīgu rediģēšanu.
   3. Ja kāds no obligātiem laukiem nav ievadīts, tad izvadīt par to kļūdas paziņojumu.
   4. Ja kāds no obligātiem laukiem neatbilst validācijas prasībām, tad izvadīt par to kļūdas paziņojumu.
10. Jānodrošina iespēju administratoram dzēst kategorijas.
    1. Pirms kategorijas dzēšanas ir jāizvada paziņojums, vai tiešām vēlaties dzēst konkrēto ierakstu.
11. Jānodrošina administratoram pievienot ražotājus.
    1. Jāparedz ieejas informācijas par preci (skat. 2.1.6. Ieejas informācijas apraksts) ievadīšana un pārbaude uz formāta pareizību.
    2. Jāparedz iespēju izvadīt paziņojumu, kad ražotājs ir veiksmīgi pievienots.
12. Jānodrošina iespēju administratoram rediģēt ražotājus.
    1. Jāparedz ieejas informācijas par ražotāju (skat. 2.1.6. Ieejas informācijas apraksts) ievadīšana un pārbaude uz formāta pareizību.
    2. Ja ražotājs ir veiksmīgi rediģēts izvadīt paziņojumu par veiksmīgu rediģēšanu.
    3. Ja kāds no obligātiem laukiem nav ievadīts, tad izvadīt par to kļūdas paziņojumu.
    4. Ja kāds no obligātiem laukiem neatbilst validācijas prasībām, tad izvadīt par to kļūdas paziņojumu.
13. Jānodrošina iespēju administratoram dzēst ražotājus.
    1. Pirms ražotāja dzēšanas ir jāizvada paziņojums, vai tiešām vēlaties dzēst konkrēto ierakstu.
14. Jānodrošina iespēju reģistrētam lietotajam pievienot, noņemt preces no groza.
    1. Jānodrošina iespēja lietotājam pievienot vai noņemt preču daudzumu.
15. Jānodrošina iespēju reģistrētam lietotājam veikt pasūtījumu.
    1. Jānodrošina lietotājam pirms pasūtījuma veikšanas iespēju apskatīties, ko viņš pasūta, kādā daudzumā.
    2. Jānodrošina lietotājam iespēju izmainīt pasūtījuma detaļas, pirms pasūtījuma veikšanas.
16. Jānodrošina reģistrētam lietotājam apskatīt savu pasūtījumu vēsturi.
    1. Jānodrošina sadaļa, kurā lietotājs var pārskatīt visus veiktos pasūtījumus, to statusus un detaļas.
    2. Lietotājam jānodrošina iespēja filtrēt pasūtījumu vēsturi pēc datuma, statusa vai citiem kritērijiem.

## Nefunkcionālās prasības

1. Sistēmas saskarnes valodai ir jābūt angļu valodai.
2. Jānodrošina tīmekļa lietojumprogrammas pielāgošanas ekrāna izmēriem, kas mūsdienās tiek lietoti, lai to varētu izmantot uz dažādiem monitora izmēriem.
3. Sistēmas dizainam ir jābūt nepārprotamam.
4. Tekstam jābūt tumšā krāsā ar viegli salasāmu fontu.
5. Sistēmai ir jāiekļauj uzņēmuma krāsu palete.
6. Sistēmai ir jābūt ātrdarbīgai.
7. Sistēma ir jābūt saderīgai ar populārākajiem tīmekļa pārlūkiem, Chrome, Firefox, Edge

*  Sistēmas galvenās lapas skice (skat. 2.3.1. att.).
  + 1. att. Sistēmas galvenās lapas skice

Šī skice attēlo sistēmas interfeisu, kuru redzēs pilnīgi visi, gan viesi, gan lietotāji, gan administratori. Skatā var redzēt reklāmas bannerus, pārdotākās preces, uzņēmuma logo, uzņēmuma kontaktinformāciju, preču sadaļas, kā arī iespēju iziet no sesijas un apskatīt savu profilu un grozu.

* Ienākšanas loga skice (skat. 2.3.2. att.)
  + 1. att. Sistēmas galvenās lapas skice

Šī skice attēlo sistēmas interfeisu, kuru redzēs pilnīgi visi, gan viesi, gan lietotāji, gan administratori. Skatā var redzēt ienākšanas logu ar laukiem priekš paroles un e-pasta.

# UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS

Elektropreču internetveikala datu uzskaites sistēma ir paredzētā izmantošanai pārlūkprogrammās, gan datoros, gan mobilajās ierīcēs. Sistēma ir sadalīta divās daļās – lietotāja daļa (frontend) un servera puse (backend). Visi dati kurus ir nepieciešams uzglabāt un organizēt tiks glabāti datubāzē.

Lietotāja daļai (frontend) tiek izmantots Vue.js[8] versija - 3.3.4 un Vite.js[7] versija - 4.4.0. Vue.js un Vite.js kombinācija tiek izmantota, jo tā piedāvā ātru un efektīvu veidu, kā izstrādāt dinamiskas lietotāja saskarnes. Vue.js ir viegli uzturams un nodrošina modulāru komponentu struktūru. Vite.js nodrošina ātru izstrādi, uzlabojot attīstības laiku. Papildus tiek izmantots Bootstrap[2] – versija 5.3, lai vieglāk būtu veidot konsekventus un responsīvus dizainus.

Lai savienotu lietotāja daļu (frontend) un servera pusi (backend) projektā tiek izmantots Axios[1] versija - 1.6.2 tādējādi ļaujot lietotājam veikt HTTP pieprasījumus uz servera pusi izmantojot API.

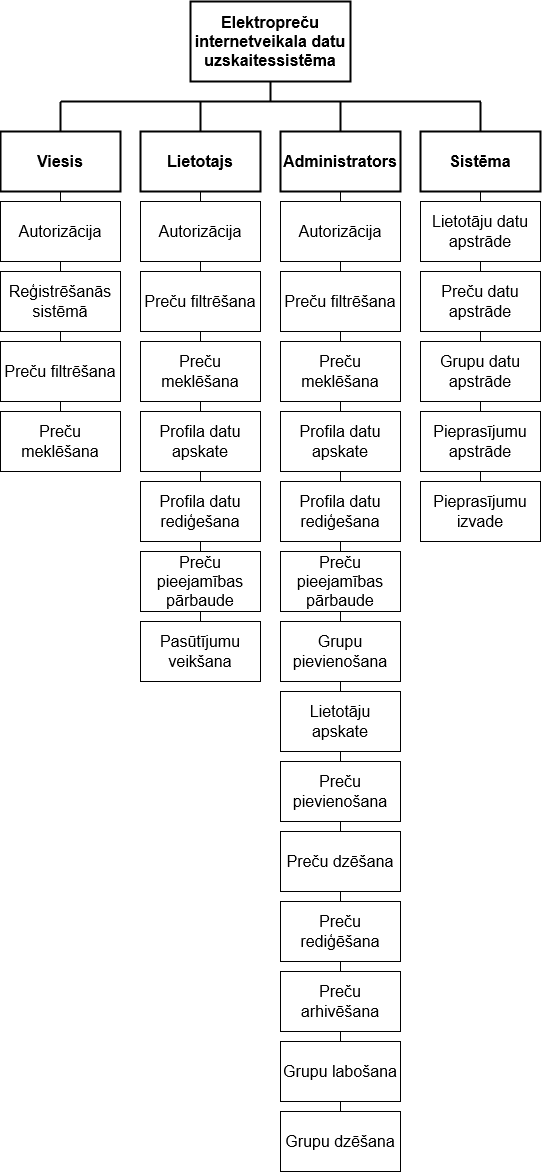
Servera pusei (backend) tiek izmantots XAMPP Control Panel versija - 3.3.0, lai varētu palaist Apache versija - 2.4.54 un MySQL[5], versija - 10.4.27-MariaDB. XAMPP tiek izmantos, jo tas ļauj viegli izveidot lokālu serveri. Datubāzes izstrādei un uzturēšanai tiek izmantots phpMyAdmin[7], jo tas ir paredzēts MySQL datubāžu pārvaldīšanai izmantojot ērtu tīmekļa saskarni. Servera puses kodam (backend) tiek izmantots PHP versija - 8.0.26 un Laravel[4] framework versija - 9.52.16, nodrošinot efektīvu un modernu veidu, kā strukturēt (backend) kodu, piedāvājot MVC (Model-View-Controller) arhitektūru.

Sistēmas iztrādei ir izmantots Visual Studio Code versija - 1.85.0. Node.js versija -18.13.0. Git[3] versija - 2.32.0.windows.2.

# PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

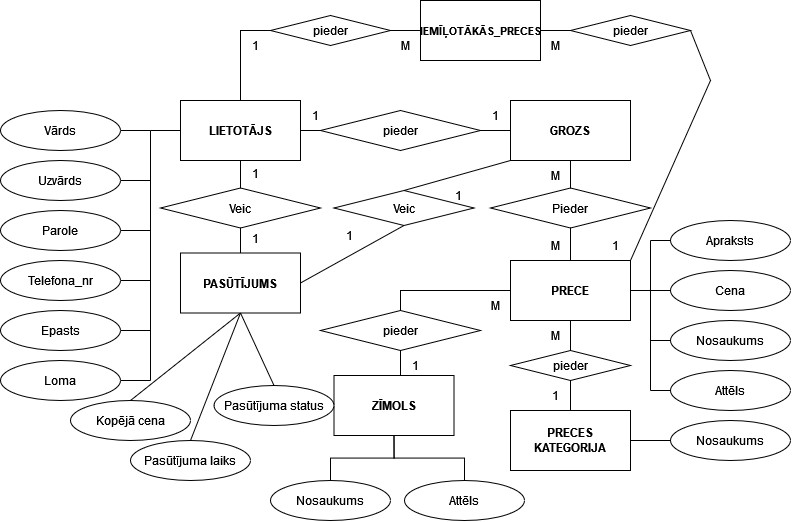
## Sistēmas struktūras modelis

### **Sistēmas arhitektūra**

Sistēmai (skat. 4.1.1.1. att.) būs četri modeļi – viesis, lietotājs, administrators, sistēma.

* + - 1. att. Funkcionālās dekompozīcijas diagramma
* **Viesu modelis.** Nereģistrētam lietotājam – viesim būs iespēja izveidot jaunus sistēmas lietotājus, kā arī apskatīt preces, kuras atrodamas internetveikalā, tās meklēt un filtrēt. Viesiem pēc jauna sistēmas lietotāja izveides būs iespēja autorizēties, tādejādi kļūstot par sistēmas lietotāju.
* **Lietotāju modulis.** Reģistrētam lietotājam būs iespēja autorizēties sistēmā. Autorizējoties sistēmā lietotājam tiks dotas papildus funkcijas, kas nav pieejamas viesim. Lietotājam būs iespēja apskatīt preces, meklēt preces, filtrēt preces. Lietotājam būs iespēja apskatīt savus profila datus, kā arī rediģēt tos, ja ir nepieciešams, kā arī lietotājam būs iespēja veikt pasūtījumu no izvēlētajām precēm.
* **Administratora modelis.** Administratoram būs visas tās pašas funkcijas, kādas būs lietotājam, bet papildus administratoram būs iespēja pārvaldīt pasūtījumus, lietotājus, pievienot preces un kategorijas, rediģēt tās, kā arī dzēst un arhivēt, ja nepieciešams.
* **Sistēmas modelis.** Sistēmas modelis būs atbildīgs par datu pieprasījumu izvadi – vizuālā daļa, ko redzēs lietotāji, viesi un administratori, kā arī datu pieprasījumu apstrādi. Kā arī sistēma atbildēs par visu datu apstrādi, to glabāšanu dzēšanu, rediģēšanu.

### **Sistēmas ER-modelis**

Sistēmas ER-modelis sastāv no 6 entītijām (skat. 4.1.2.1. att.), kas nodrošina pamat informācijas uzglabāšanu un apstrādi. Tie ir Lietotājs, Prece, Preces kategorija, Grozs, Pasūtījums, Zīmoli.

* + - 1. att. Sistēmas ER-diagramma
* **Lietotājs** – apraksta lietotāju. Atribūtu kopums sevī ietver lietotajā vārdu, uzvārdu, paroli, telefona numuru, e-pastu, lomu.
* **Prece** – apraksta preci. Atribūtu kopums sevī ietver preces nosaukumu, preces aprakstu un tās cenu.
* **Preces kategorija** - apraksta preces kategorija. Atribūtu kopums sevī ietver preces kategorijas nosaukumu.
* **Pasūtījums** – apraksta pasūtījumu. Atribūtu kopums sevī ietver pasūtījuma laiku.
* **Grozs** – apraksta preču grozu. Atribūtu kopums sevī ietver preču daudzumu un kopējo cenu.
* **Zīmols** – apraksta zīmolu. Atribūtu kopums sevī ietver zīmola nosaukumu un attēlu.

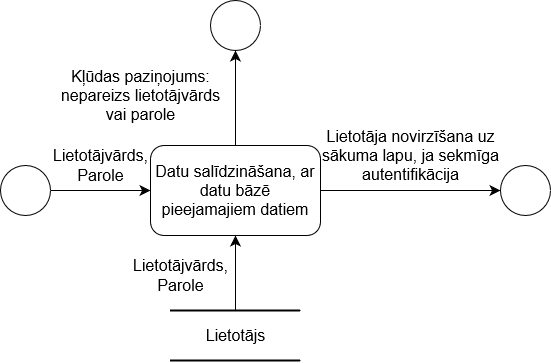
Datu bāzes relācijas uzrāda kā savstarpēji ir savienotas divas entītijas.

* Starp **Prece** un **Preces kategorija** attiecība ir viens pret daudziem, jo vienai kategorijai var būt vairākas preces, bet viena prece var piederēt tikai pie vienas preces kategorijas.
* Starp **Prece** un **Zīmols** attiecība ir viens pret daudziem, jo vienam zīmolam var būt vairākas preces, bet vienai precei var būt tikai viens zīmols.
* Starp **Prece** un **Grozs** attiecība ir daudzi pret daudziem, jo vairākas preces var būt vairākos grozos.
* Starp **Lietotājs** un **Grozs** attiecība ir viens pret viens, jo vienam lietotājam var būt tikai viens grozs
* Starp **Lietotājs** un **Pasūtījums** attiecības ir viens pret daudziem, jo viens lietotājs var veikt vairākus pasūtījumus.
* Starp **Grozs** un **Pasūtījums** attiecība ir viens pret vienu, jo viens pasūtījums var izmantot tikai vienu grozu.

## Funkcionālais sistēmas modelis

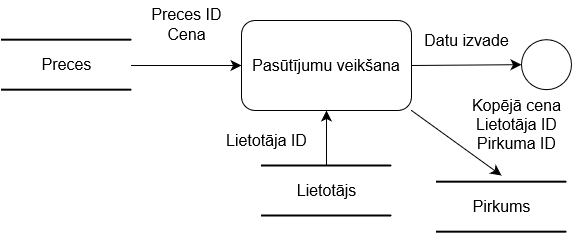
### **Datu plūsmu modelis**

1. **Lietotāju autorizēšanās**

Autorizācijas modelis (skat. 4.2.1.1. att.) atbildēs par lietotāja autorizāciju sistēmā. Lietotājam ir jāievada e-pasts un parole. Ja lietotājs ir ievadījis pareizus datus, tad tiks ielaists sistēmā, kā konkrētais lietotājs. Ja dati būs ievadīti nekorekti, tad tiks izvadīts paziņojums, ka e-pasts vai parole ir nepareizs.

* + - 1. att. Lietotāja autorizācijas datu plūsmu diagramma

1. **Lietotāja reģistrēšana datu bāzē**

**** Reģistrēšanas modelis (skat. 4.2.1.2. att.) atbildēs par lietotāja reģistrēšanu datu bāzē. Viesim būs jāizpilda ailes, kurās jāievada e-pasts, vārds, uzvārds, parole, parole atkārtoti un telefona numurs. Ja visi lauki būs aizpildīti un tie atbildīs visām validācijas prasībām, tad tiks izveidots jauns lietotāja profils un būs iespēja reģistrēties sistēmā. Ja kāda no validācijas prasībām nebūs veiksmīga, tad viesim tiks izvadīts kļūdas paziņojums – kura no ailēm neatbilst prasībām.

* + - 1. att. Lietotāja reģistrēšanas datu plūsmu diagramma

1. **Profila datu izvade**

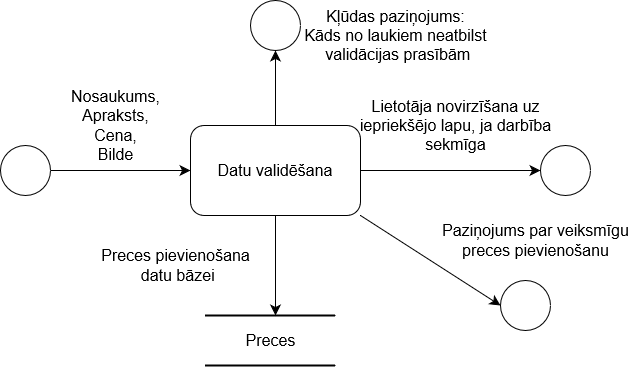
Profilu datu izvade atbildēs par konkrētā reģistrētā lietotāja datu izvadi, šos datu redzēs tikai un vienīgi profila īpašnieks. Profila īpašniekam par sevi būs iespējams aplūkot šādus datus – vārds, uzvārds, e-pasts, telefona numurs.

1. **Lietotāja profila datu rediģēšana**

Lietotāja profila datu rediģēšana atbildēs par konkrēta lietotāja datu rediģēšanu. Lietotāju datu rediģēšana sadarbojas ar profila datu izvadi. Ja lietotājs pamana, ka dati ir nekorekti vai mainījušies, tam būs iespēja rediģēt šos datus. Ja datu validācija būs sekmīga, tad lietotāja dati tiks rediģēti datu bāzē. Ja validācija būs neveiksmīga, tad tiks izvadīts kļūdas paziņojums – kura no ailēm neatbilst prasībām.

1. **Preču ierakstu pievienošana datu bāzē**

Preču ierakstu pievienošana (skat. 4.2.1.3. att.) atbildēs par preces pievienošanu datu bāzei. Administratoram būs iespēja pievienot jaunu preci sistēmai aizpildot ievades laukus. Lai pievienotu jaunu preci ir nepieciešams ievadīt preces nosaukumu, aprakstu, cenu, attēlu. Ja datu validācija būs sekmīga, tad prece tiks pievienota datu bāzei. Ja kāda no validācijas prasībām nebūs veiksmīga, tad tiks izvadīts kļūdas paziņojums – kura no ailēm neatbilst prasībām un prece netiks pievienota datu bāzei.

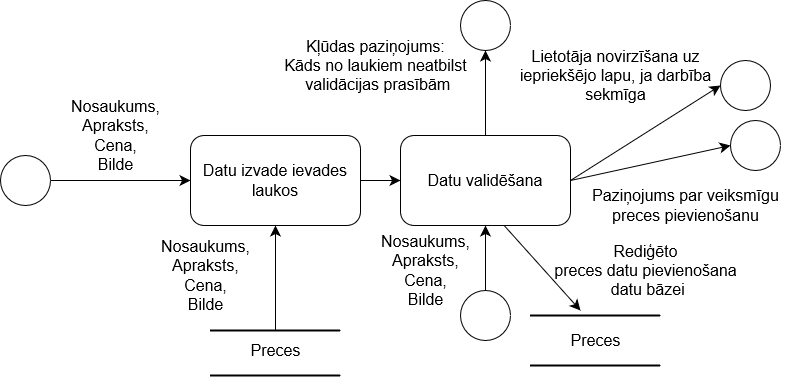


* + - 1. att. Preču pievienošanas datu plūsmu diagramma

1. **Ierakstu meklēšana**

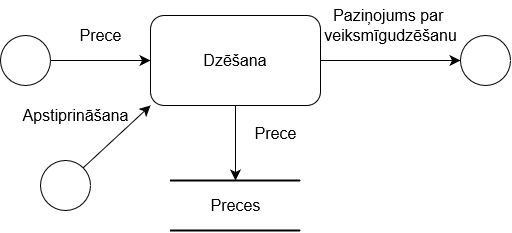
Ierakstu meklēšana atbildēs par preču meklēšanu pēc specifiskas frāzes vai vārda vai burtiem. Speciālā meklēšanas laukā lietotājam būs jāievada piemēram konkrēts preces nosaukums, pēc tam pēc šīs frāzes tiks meklēts, vai atbilst kāds ieraksts, ja ieraksts datubāzē tiks atrasts, tad tas tiks izvadīts, ja ieraksts netiks atrasts tiks izvadīts paziņojums, ka neviena prece neatbilst meklētajai.

1. **Preču ierakstu rediģēšana datu bāzē**

Preču datu rediģēšana (skat. 4.2.1.4. att.) atbildēs par konkrētas izvēlētās preces datu rediģēšanu. Administratoram būs nepieciešams attiecīgajās ailēs, kurās tas būs nepieciešams izmainīt datus piemēram cenu. Pēc datu rediģēšanas dati izies cauri validācijai, ja viss atbildīs prasībām, tad dati tiks veiksmīgi rediģēti datubāzē, kā arī datubāzē uzrādīsies, kad dati ir rediģēti. Ja validācija būs neveiksmīga, tad tiks izvadīts kļūdas paziņojums – kura no ailēm neatbilst prasībām.

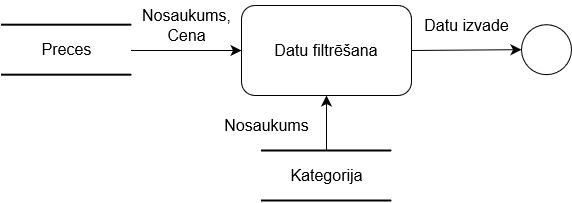
* + - 1. att. Preču rediģēšanas datu plūsmu diagramma

1. **Preču ierakstu dzēšana datu bāzē**

Preču ierakstu dzēšana (skat. 4.2.1.5. att.) atbildēs par konkrētās izvēlētās preces dzēšanu no datu bāzes ierakstiem. Administratoram būs jāizvēlas, kuru preci vēlas dzēs, pēc tā tiks izvadīts paziņojums, vai tiešām vēlaties dzēst preci, ja tiks nospiests jā, tad ieraksts tiks dzēst no datu bāzes, ja lapa tiks aizvērta vai uzspiests nē, tad nekas nenotiks.

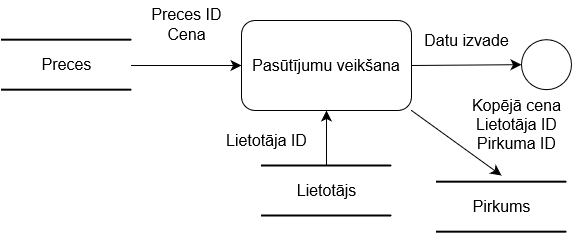
* + - 1. att. Preču dzēšanas datu plūsmu diagramma

1. **Ierakstu filtrēšana**

Ierakstu filtrēšana (skat. 4.2.1.6. att.) atbildēs par preču filtrēšanu, būs iespēja izvēlēties, kā kārot preces piemēram cena augoša vai cena dilstoša, cena no līdz. Lietotājam ievadot kādu no filtrēšanas opcijām dati tiks attiecīgi filtrēti un izvadīti tikai filtriem atbilstošie dati. Ja iestatot filtrus neviena prece neatbildīs prasībām, tad tiks izvadīts neviena prece netika atrasta.

* + - 1. att. Ierakstu filtrēšanas datu plūsmu diagramma

1. **Pasūtījumu veikšana**

Pasūtījumu veikšana (skat. 4.2.1.7. att.) atbildēs par gala pasūtījuma veikšanu. Lietotājam veicot pasūtījumu tiks prasīts apstiprināt izvēlētās preces un to gala cenu, ja lietotājs izvēlas doties tālāk, tad tiks izveidots pasūtījums un lietotājam būs iespēja izrentēt pasūtījuma dokumentu. Ja lietotājs neizvēlas turpināt pasūtījuma, tad tas netiek izveidots un lietotājs tiek novirzīts atpakaļ uz iepriekšējo lapu.

* + - 1. att. Pasūtījumu veikšanas datu plūsmu diagramma

# DATU STRUKTŪRU APRAKSTS

Datubāze sastāv no 6 tabulām (skat. 5.1. att.), kas satur sevī informāciju par lietotājiem, precēm, kategorijām, grozu, pasūtījumu un groza objektiem.

* 1. att. Tabulu relāciju shēma

1. Tabula **“user”** glabā datus par lietotājiem.
2. Tabula **“item”** glabā datus par precēm.
3. Tabula **“category”** glabā datus par kategorijām.
4. Tabula **“purchases”** glabā datus par pasūtījumiem.
5. Tabula **“cart”** glabā datus par precēm, kurus lietotājs ielicis grozā.
6. Tabula **“brands”** glabā datus par zīmoliem**.**
7. Tabula **“favorite\_items”** glabā datus par iecienītākajām lietotāja precēm.

Tabula “**user**” glabā datus par reģistrētiem lietotājiem. Tabulas “**user**” primārā atslēga ir “user\_id”, tabula ir saistīta ar tabulu “**cart**”.

* 1. tabula

Tabulas “**user**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | user\_id | bigint | 20 | Ieraksta identifikators, primārā atslēga (unikāls) |
| 2. | name | varchar | 255 | Lietotajā vārds |
| 3. | surname | varchar | 255 | Lietotāja uzvārds |
| 4. | password | varchar | 255 | Šifrēta parole, ko lietotājs ir izvēlējies |
| 5. | phone | int | 11 | Lietotāja teflona numurs |
| 6. | email | varchar | 255 | Lietotāja e-pats |
| 7. | admin | tinyint | 1 | Lietotāja loma |
| 8. | created\_at | timestamp | - | Kad tika izveidots lietotājs |
| 9. | updated\_at | timestamp | - | Kad tika atjaunoti lietotāja dati |

Tabula “**category**” glabā datus par preču kategorijām, tabulai ir unikāla primārā atslēga “category \_id”. Tabula “**category**” ir saistīta ar tabulu “**item**”.

* 1. tabula

Tabulas “**category**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | category \_id | bigint | 20 | Ieraksta identifikators, primārā atslēga (unikāls) |
| 2. | name | varchar | 255 | Kategorijas nosaukums |
| 4. | created\_at | timestamp | - | Kad tika izveidots lietotājs |
| 5. | updated\_at | timestamp | - | Kad tika atjaunoti lietotāja dati |

Tabula “**item**” glabā datus par precēm, tabula ir saistīta ar trim citām “**category**”, “**basket\_item**” un “**brands**”. Tabulas “**item**” primārā atslēga ir “item\_id”, kas apraksta katras preces unikālo identifikatoru, kā arī tabulai ir ārējās atslēgas “category\_id” un “brands\_id”, kas atsaucas uz tabulām “**category**” un “**brands**”.

* 1. tabula

Tabulas “**item**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | item\_id | bigint | 20 | Ieraksta identifikators, primārā atslēga (unikāls) |
| 2. | name | varchar | 255 | Preces nosaukums |
| 3. | desc | varchar | 255 | Preces apraksts |
| 4 | price | float | - | Preces cena |
| 5. | img | varchar | 255 | Tiek glabāts preces attēla ceļš -EVk7LOgI2G.png |
| 6. | created\_at | timestamp | - | Kad tika izveidots lietotājs |
| 7. | updated\_at | timestamp | - | Kad tika atjaunoti lietotāja dati |
| 8. | brands\_id | bigint | 20 | Ārējā atslēga |
| 9.. | category\_id | bigint | 20 | Ārējā atslēga |

Tabula “**favorite\_items**” glabā datus par iecienītākajām lietotāja precēm, tabulai ir unikāla primārā atslēga “category \_id”. Tabula “**category**” ir saistīta ar tabulām “**item**” un ”**users**”, tabulai ir arējā atslēga “user\_id”, kas atsaucas uz tabulu ”users” un“item\_id”, kas atsaucas uz tabulu “item”.

* 1. tabula

Tabulas “**favorite\_items**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | favorite \_id | bigint | 20 | Ieraksta identifikators, primārā atslēga (unikāls) |
| 2. | item\_id | bigint | 20 | Ārējā atslēga |
| 3. | user\_id | bigint | 20 | Ārējā atslēga |

5.4. tabulas turpinājums

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4. | created\_at | timestamp | - | Kad tika izveidots lietotājs |
| 5. | updated\_at | timestamp | - | Kad tika atjaunoti lietotāja dati |

Tabula “**cart**” glabā datus par lietotāja pirkuma grozu. Tabulai ir unikāla identifikācijas atslēga “basket\_id”, kas arī ir primārā atslēga. Tabula “**cart**” ir saistīta ar tabulām “**item**” un ”**users**”, tabulai ir arējā atslēga “user\_id”, kas atsaucas uz tabulu ”users” un“item\_id”, kas atsaucas uz tabulu “item”.

* 1. tabula

Tabulas “**cart**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | cart\_id | bigint | 20 | Ieraksta identifikators, primārā atslēga (unikāls) |
| 2. | user\_id | bigint | 20 | Ārējā atslēga |
| 3. | Item\_id | bigint | 20 | Ārējā atslēga |
| 4. | created\_at | timestamp | - | Kad tika izveidots lietotājs |
| 5. | updated\_at | timestamp | - | Kad tika atjaunoti lietotāja dati |

Tabula “**purchases**” glabā datus par pasūtījumiem. Tabula ir saistīta ar tabulām “**item**” un “**users**”. Tabula “**purchases**” izmanto ārējās atslēgas “itemt\_id” un “user\_id”, lai izveidotu relācijas starp tabulām “**item**” un “**users**” tabulām.

* 1. tabula

Tabulas “**purchases**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | purchases\_id | bigint | 20 | Ieraksta identifikators, primārā atslēga (unikāls) |
| 2. | status | varchar | 255 | Pasūtījuma status active vai closed |
| 3 | total\_price | decima | 8,2 | Kopējā cena, kas ir jāmaksā |

5.6. tabulas turpinājums

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4. | item\_id | bigint | 20 | Ārējā atslēga |
| 5. | user\_id | bigint | 20 | Ārējā atslēga |
| 6. | created\_at | timestamp | - | Kad tika izveidots lietotājs |
| 7. | updated\_at | timestamp | - | Kad tika atjaunoti lietotāja dati |

Tabula “**brands**” glabā datus par zīmoliem. Tabulai ir unikāla identifikācijas atslēga “brands\_id”, kas arī ir primārā atslēga. Tabula “**brands**” nav nav nevienas ārējās atslēgas.

* 1. tabula

Tabulas “**brands**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | brands\_id | bigint | 20 | Ieraksta identifikators, primārā atslēga (unikāls) |
| 2. | name | varchar | 255 | Zīmola nosaukums |
| 3. | img | varchar | 255 | Tiek glabāts preces attēla ceļš -EVkLOgI2G.png |
| 4. | created\_at | timestamp | - | Kad tika izveidots lietotājs |
| 5. | updated\_at | timestamp | - | Kad tika atjaunoti lietotāja dati |

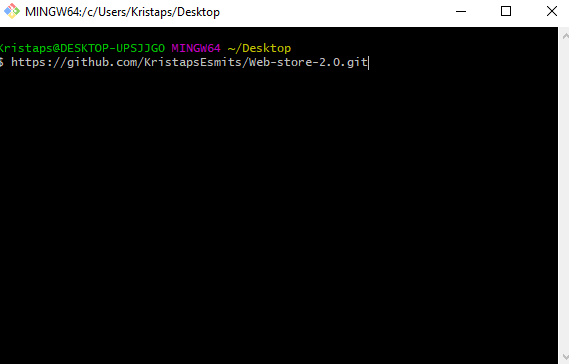
# LIETOTĀJA CEĻVEDIS

## Sistēmas prasības aparatūrai un programmatūrai

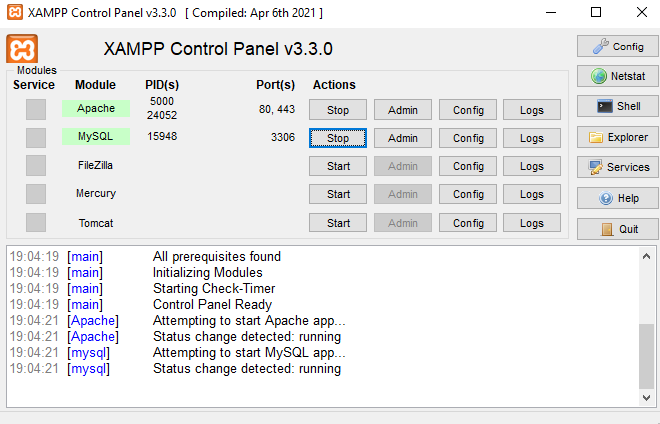
Sistēma tika realizēta kā interneta timekļa vietne, kuras lietošana neprasa specifisku programmu instalāciju, kā arī var tikt palaista no jebkuras ierīces, kurai ir:

* Viena no jaunākajām pārlūkprogrammu versijām
* Safari – 13. versija un jaunāka;
* Windows Edge – 100.0 un jaunāk;
* Google Chrome – 100.0 un jaunāka;
* Opera – 84.0.4316.31 un jaunāka;
* Mozila Firefox – 100.0 un jaunāka;
* Stabils interneta savienojums – jo labāks savienojums, jo veiksmīgāk noritēs programmas darbība;
* Aparatūra, lai uz tās varētu darbināt iepriekš šajā sadaļā minētās pārlūkprogrammas.

## Sistēmas instalācija un palaišana

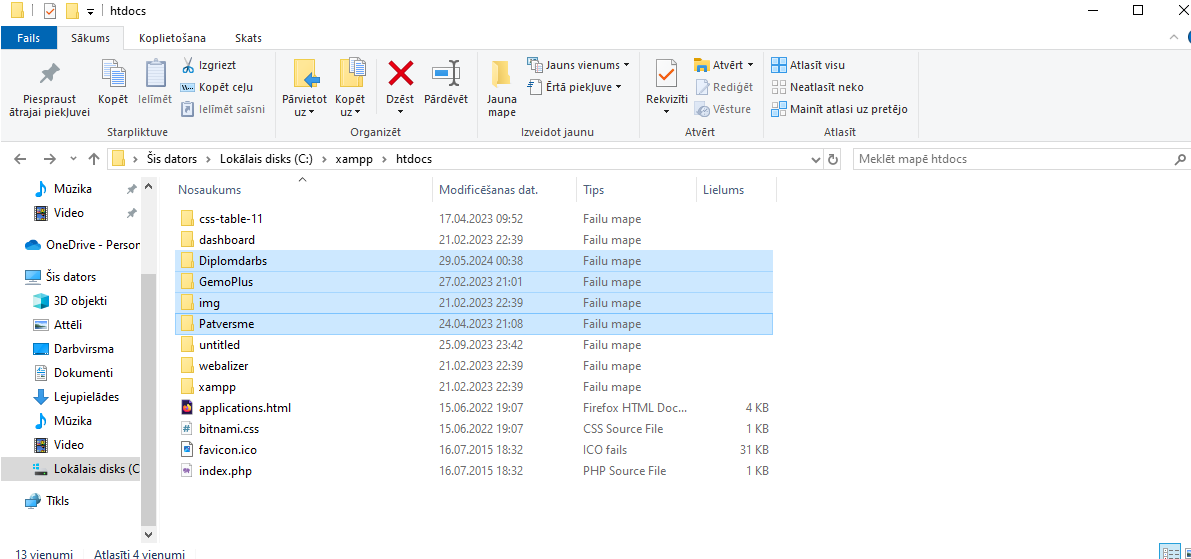
Sistēmas instalācijai un palaišanai būs nepieciešams papildus instalēt xamp, git. Nepieciešams instalēt git, lai būtu iespējams noklonēt projektu no Github. Git var lejupielādēt https://git-scm.com/downloads. Jāizvēlas atbilstošā datora operētājsistēma un versija – portable. Ielādējot programmu un atverot to padīdīsies konsoles logs, kurā jāievada projekta github links – “https://github.com/KristapsEsmits/Web-store-2.0.git” (skat. 6.2.1. att.).

* + 1. att. Git konsoles logs

Uzspiežot enter taustiņu tiks lejupielādēts projekts. Lai projektu palaistu ir nepieciešams instalēt XAMP[9] – “https://www.apachefriends.org/download.html” lietotājs lejupielādē attiiecīgi sev nepieciešamo versiju. Palaižot programmu lietotājam jāpalaiž Apache server un MySql server (skat. 6.2.2. att.). Palaist Apache server un MySql serveri n nepieciešams katru reizi ieslēdzot sistēmu, Apache server un MySql server atbild par backend un datubāzes darbību

* + 1. att. Xamp programmas logs

Tālāk jāpārkopē vai jāpārliek lejupielādētais Github projekts XAMP mapē - xampp\htdocs (skat. 6.2.3. att.)., ja tas netiks izdarīts XAMP nespēs atrast projektu un datubāze nekomunicēs savā starpā ar backend un otrādi.

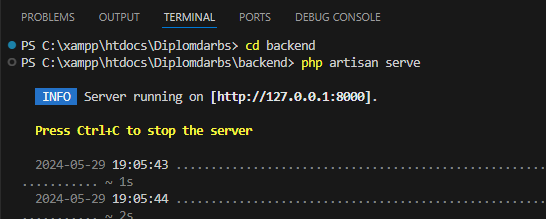


* + 1. att. Piemērs, kur jāiekopē novilktais projekts

Tālāk jāatver projekts konsolē, vai koda redaktora konsolē un jāievada sekojošas komandas (skat. 6.2.4. att.).:

\* Ievadot šīs komandas tiek palaists projekta backend

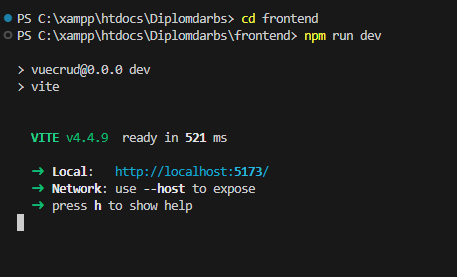
* cd backend
* php artisan serve



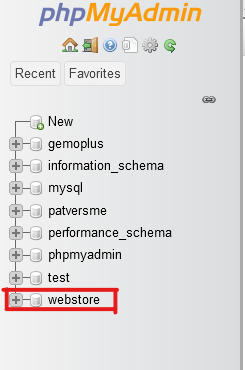
* + 1. att. Konsole ar ievadītām komandām priekš backend palaišanas

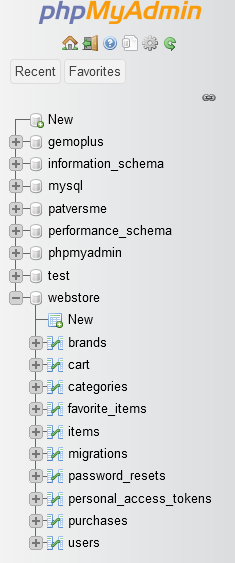
Tālāk jāatver jauna konsole neaizverot ciet esošo konsoli un jāievada sekojošas komandas (skat. 6.2.5. att.).:

\* Ievadot šīs komandas tiek palaists projekta frontend

* cd frontend
* npm run dev
  + 1. att. Konsole ar ievadītām komandām priekš frontend palaišanas

Sekmīgi veicot visas iepriekš minētās darbības jābūt iespējai atvērt projektu lokāli, to var izdarīt spiečot uz linka, kas tiek izvadīts konsolē (skat. 6.2.5. att.). Lai piekļūtu datubāzei vai veiktu manipulācijas ar to nepieciešams atvērt linku: <http://localhost/phpmyadmin/> un izveidot jaunu datubāzi ar nosaukumu webstore (skat. 6.2.6. att.). Tālāk jāatver projekts jaunā konsolē un jāievada sekojošas komandas:

* cd backend
* php artisan migrate
  + 1. att. Attēls ar izveidotu jaunu datubāzi

 Pēc veiksmīgas migrācijas jāparādās visām nepieciešamajām tabulām zem datubāzes webstore (skat. 6.2.7. att.).

* + 1. att. Attēls ar izveidotām tabulām datubāzē webstore

## Programmas apraksts

## Testa piemērs

* 1. tabula

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr.** | **Prasības numurs** | **Prasības nosaukums** | **Ievaddati/situācijas apraksts** | **Sagaidāmais rezultāts** | **Statuss** |
| 1. | 1 | Jauna lietotāja reģistrēšana | Neievadot nekādu informāciju tiek nospiesta poga “Register” | Tiks izvadīta informācija:The name field is required.  The surname field is required.  The password field is required.  The phone field is required.  The email field is required. | Pareizi |
| 2. | Tiek ievadīti burti telefona numura vietā, netiek ievadīts e-pasts un paroles nesakrīt | Tiks izvadīta informācija: The password confirmation does not match.  The phone must be a number.  The email must be a valid email address. | Pareizi |
| 3. | Visi dati tiek ievadīti korekti un nospiesta poga register | Tiks izvadīts paziņojums “User registered successfully!” | Pareizi |

6.1. tabulas turpinājums

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4. | 7 | Preces dzēšana | Tiek uzspiesta poga delete items | Parādās PopUp ar teksu “Are you sure?” | Pareizi |
| 5. | Pēc PopUp tiek uzspiests “cancel” | Prece netiek izdzēsta | Pareizi |
| 6. | Pēc PopUp tiek nospiest “delete” | Prece tiek izdzēsta un lietotājam par to tiek paziņots | Pareizi |
| 7. | 2 | Autorizēšanās | Tiek ievadīti pareizi dati: epasts un parole | Lieottāja ievadītie dati tiek salīdzināti ar datu bāze esošajiem datiem, tie sakrīt un lietotājs tiek ielaists savā profilā | Pareizi |
| 8. | Lietotājvārda vietā ievada nepareizu e-pastu | Tieks izvadīts paziņojums “Incorrect email or password. Please try again” | Pareizi |
| 9. | Netiek ievadīts nekas un uzspiests login | Tieks izvadīts paziņojums “Incorrect email or password. Please try again” | Pareizi |
| 10. | 4.1 | Preču filtēšana | Tik ievadīta min un max cena | Tiks izvadītas tikai tās preces, kuras ietilpst šajā preču kategorijā | Pareizi |
| 11. | Ievadīti filti, kuri nesakrīt ar nevienu preci | Tiks izvadīts paziņojums, ka prece ar šādiem kritērijiem netika atrasta | Pareizi |
| 12. | Tiek noņemti visi filti | Lietotājam tiks izvadītas visas preces | Pareizi |

# NOBEIGUMS

Kvalifikācijas darba izstrādes laikā tika izveidota “Elektropreču internetveikala datu uzskaites sistēma”, kas nodrošina iespēju jebkuram lietotājam, neatkarīgi no viņa tehniskās sagatavotības, viegli un ērti pārvaldīt veikala datus. Sistēmas backend izstrādei tika izmantots PHP ar Laravel ietvaru un Axios, savukārt frontend izstrādei tika izvēlēts Vue.js, Vite.js un Bootstrap. Datu glabāšanai tika izmantota MySQL datubāze ar phpMyAdmin, un sistēma tika izvietota XAMPP serverī, izmantojot Apache. Sistēmas funkcionalitāte ietver preču pievienošanu, labošanu, dzēšanu, pasūtījumu veikšanu un preču meklēšanu, kā arī lietotāju reģistrāciju un autorizāciju.

Pašlaik sistēma tiek izmantota personiskām vajadzībām, bet to var izmantot jebkurš, kuram ir vēlme, jo sistēma ir opensource. Sistēmu noteikti varēs attīstīt uz kaut ko lielāku, pašlaik ir plāns pārveidot visu frontend uz Angular 17, tādējādi uzlabojot lietotaju pieredzi un sistēmas atrdarbību. Sistēmai arī ir paradzētas backend izmaiņas uzlabojot arhitektūru, tādējadi samazinot api pieprasījumu skaitu uzlabojot sistēmas ātrdarbību.

# INFORMĀCIJAS AVOTI

1. [ANG] Axios dokumentācija - <https://axios-http.com/docs/intro> - (Resurss apskatīts 04.12.2023.).
2. [ANG] Bootstrap dokumentācija - <https://getbootstrap.com/> - (Resurss apskatīts 28.11.2023.).
3. [ANG] Git dokumentācija - <https://git-scm.com/docs/gitignore> - (Resurss apskatīts 23.11.2023.).
4. [ANG] Laravel dokumentācijā - <https://laravel.com/docs/10.x> - (Resurss apskatīts 06.11.2023.).
5. [ANG] MySql dokumentācija - <https://dev.mysql.com/doc/> - (Resurss apskatīts 29.05.2024).
6. [ANG] PhpMyAdmin dokumentācija - <https://www.phpmyadmin.net/docs/> - (Resurss apskatīts 29.05.2024).
7. [ANG] Vite.js dokumentācijā - <https://vitejs.dev/> - (Resurss apskatīts 04.12.2023.).
8. [ANG] Vue.js dokumentācijā - <https://vuejs.org/guide/introduction.html> - (Resurss apskatīts 06.11.2023.).
9. [ANG] XAMP dokumentācija - <https://www.apachefriends.org/> - (Resurss apskatīts 29.05.2024).

# PIELIKUMI

## 1.pielikums. Pārskatu piemēri

## 2.pielikums. Ekrānattēlu piemēri

## 3.pielikums. Programmas pirmteksts