

Profesionālās izglītības kompetences centrs

“Rīgas Valsts tehnikums”

Izglītības programma: Programmēšana

KVALIFIKĀCIJAS DARBS

Interneta veikala datu uzskaites automatizētā sistēma

Paskaidrojošais raksts: ? lpp

Audzēknis:

Deniss Štrombergs

Vadītājs:

Igors Litvjakovs

Normu kontrole:

Olga Sabanska

Rīga

2021

SATURS

ANOTĀCIJA.....	4
ANNOTATION.....	5
IEVADS.....	6
1 UZDEVUMA NOSTĀDNE.....	6
2 PRASĪBU SPECIFIKĀCIJA.....	7
2.1 Ieejas un izejas informācijas apraksts.....	7
2.1.1 Ieejas informācijas apraksts.....	7
2.1.2 Izejas informācijas apraksts.....	9
2.2 Funkcionālās.....	9
2.4 Nefunkcionālās.....	9
3 UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS.	9
4 PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN	
PROJEKTĒŠANA.....	11
4.1 Sistēmas arhitektūra.....	11
4.1.1 Administratora daļa.....	11
4.1.2 Klienta daļa.....	11
4.2 Sistēmas ER modelis.....	11
4.3 Funkcionālais sistēmas modelis.....	13
5 DATU STRUKTŪRAS APRAKSTS.....	14
6 LIETOTĀJA CEĻVEDI.....	21
6.1 Sistēmas prasības aparatūrai un programmatūrai.....	21
6.1.1 Windows.....	21
6.1.2 Linux (Ubuntu).....	22
6.2 Sistēmas instalācija un palaišana.....	22
6.3 Programmas apraksts.....	23
6.3.1 Administrators.....	23
6.3.2 Klients.....	24
6.4 Testa piemērs.....	24

NOBEIGUMS.....	25
INFORMĀCIJAS AVOTI.....	26
PIELIKUMI.....	27
ER diagramma.....	27
Fiziska struktūra.....	28
Sistēmas arhitektūra.....	29

ANOTĀCIJA

Kvalifikācijas darbā ir aprakstīta interneta veikala automatizētas sistēmas izstrādāšanas process. Sistēma dod iespēju tā lietotājam, apkopot un apstrādāt datus par sava veikala produktiem un klientiem, konkrēti, var veidot produktus, kuri atspoguļo reālus produktus, norādīt produkta atribūtus (cena, krāsa u.c.), kā arī, var redzēt kurus produktus iegādājās pircēji, un kādas atsauksmes tie atstāj. Sistēma tika izstrādāta ar JavaScript programmēšanas valodu, izmantojot React ietvaru, lai realizētu "Frontend", un "Node" ar "PostgreSQL", "GraphQL" un "Express", lai realizētu "Backend".

Kvalifikācijas darbs ietver ievadu, uzdevumu nostādni, prasību specifikāciju, uzdevuma risināšanas līdzekļu izvēles pamatojumu, programmatūras produkta modelēšanas un projektēšanas aprakstu, datu struktūru aprakstu, lietotāju ceļvedi, nobeigumu un pielikumus.

Kvalifikācijas darba ievadā ir aprakstīta vairumtirdzniecības aktuālā problēma un tās risinājumus. Uzdevumu nostādnē ir norādīti uzdevumi, kurus sistēmai būs nepieciešams veikt. Prasību specifikācija sastāv no ieejas un izejas informācijas, kā arī no sistēmas funkcionālajām un nefunkcionālajām prasībām. Uzdevuma risināšanas līdzekļu izvēles pamatojumā ir norādīti, kādi līdzekļi tiks izmantoti izstrādāšanai un kādiem nolūkiem tie tiek izmantoti. Programmatūras produkta modelēšanas un projektēšanas apraksts sastāv no sistēmas struktūras modeļa, kas ietver sistēmas arhitektūru un ER modeli, un funkcionālās sistēmas modeļa, kas satur datu plūsmu modeli. Datu struktūru aprakstā tiek parādīta datu bāzes relāciju shēma, kā arī tabulu struktūra ar aprakstu, kas ietver datu tipu un datu garumu norādīšanu. Lietotāja ceļvedī ir norādītas nepieciešamās sistēmas prasības aparatūrai un programmatūrai, sistēmas instalācija un palaišana, kā arī programmas apraksts, kas paskaidro, kā pareizi lietot sistēmu. Testa piemērā ir dots detalizēt produktu pievienošanas apraksts.

Kvalifikācijas darbs sastāv no ? lappusēm, kurā ietilpst ? attēli, ? tabulas un ? pielikumi. Pielikumi satur sadaļas: ER diagrammu, datu bāzes relāciju shēmu un datu arhitektūru

ANNOTATION

This qualification project contains the description of online store automatized system development process. The system will give opportunity to user to collect and process the data about their stores products and clients, to be more specific, it is possible to create products, which represent real material entities from the real word, specify product attributes (price, color, etc.). Also it is possible to view which products are bought and how much exactly, and see what kind of review customers write. The system is built using JavaScript with React framework for frontend and Node with PostgreSQL, GraphQL and Express library, to realize backend.

The introduction has been written about the shop itself, why it is useful and why people would like to use it in their daily life. Assignment approach is described in great details about the purpose and requirements for the selected task. The requirement specification will include the functionality of an online store, the possible actions to be taken both from the user's side. as well as data from the accounting system.

Task solving of resources section has been described about technology that was used in creating of this project and why this technology was chosen. Software product modeling and design system architecture is referred to the creation of the model as well as the entity relationship diagram and data flow models. Data Structures chapter is referred to the data table field types and the links between other tables. User's guide is a detailed and well-written user instructions, so the user may be aware of all the records for the system options available and to work with the online store avert mental rights difficulties and uncertainties.

Overall qualification workload is ? pages., which includes ? images, ? tables and ? attachments.

IEVADS

Kvalifikācijas darba mērķis ir izveidot interneta veikala sistēmu, līdzīgi jau eksistējošiem variantiem, kā Magento 2, Shopify, BigCommerce un citiem, lai varētu nodrošināt varētu apkopot sava veikala datus vienā sistēmā un, lai paplašinātu pircēju loku un pieejamību balstoties uz to, kā veikals ir arī pieejams internetā, kas joprojām ir ļoti aktuāli. Protams nav jēgās būvēt sistēmu, kura spēs kļūt par konkurent ar iepriekš minētām, jau eksistējošām sistēmām. Bet tas ar ko šī interneta veikala sistēma atšķirsies ir ar to, ka tā vienkāršāka. Šīs sistēmas nodrošina visas nepieciešamas bāzes funkcionalitātes, kuras varētu būt nepieciešamas, taču dažādas integrācijas ir neobligātas un nav pēc nosacījuma implementētas, piemēram, Payment Gateway, vai Shipping API, nebūs pieejami šajā programmā, vismaz pagaidām, tām vajag būt implementētam, bāzējoties uz konkrēta gadījuma nepieciešamībām. Īsumā, šī sistēma nodrošina pamata funkcionalitāti, taču specifiskas lietas, jau jābūt implementētam balstoties uz konkrētu gadījumu.

1 UZDEVUMA NOSTĀDNE

Kvalifikācijas darba uzdevums ir izveidot interneta veikala sistēmu. Sistēma sastāvēs no 2 galvenām daļām administrator un klients. Administrators, ir kā interneta veikala īpašnieks, tam ir pieeja visiem nepieciešamiem datiem un tas var to manipulēt. Klient, ir kā pircējs, tam ir pieeja veikala priekšpusei, tas var likt pirkumus grozā, un veikt pasūtījumus.

Interneta veikala sistēmas sistēmai ir jānodrošina sekojošās funkcionalitātes:

1. Administrators:

- Dažādu datu (produkti, kategorijas, lapas, atribūti, u.c.), apkopošana un manipulācija
- Kopējās veikala konfigurācijas uzstādīšana
- ...

1. Klients:

- Produktu meklēšana
- Produkta lapas apskatīšana un iespēja pievienot grozā
- Atsauksmju sistēma

- Pircēja akaunta veidošana un rediģēšana
- ...

2 PRASĪBU SPECIFIKĀCIJA

2.1 Ieejas un izejas informācijas apraksts

2.1.1 *Ieejas informācijas apraksts*

Sistēmā tiks nodrošināta šāda ieejas informācijas apstrāde. Informācijas par lietotājiem sastāvēs no sekojošiem datiem:

1 Adrese

- 1.1 Vārds – obligāts teksts ar izmēru līdz 255 rakstzīmēm
- 1.2 Uzvārds – obligāts teksts ar izmēru līdz 255 rakstzīmēm
- 1.3 Telefona numurs – obligāts teksts ar izmēru līdz 255 rakstzīmēm
- 1.4 Valsts – obligāts teksts ar izmēru līdz 255 rakstzīmēm
- 1.5 Pilsēta – obligāts teksts ar izmēru līdz 255 rakstzīmēm
- 1.6 Reģions – obligāts teksts ar izmēru līdz 255 rakstzīmēm
- 1.7 Pirmā Ielas līnija – obligāts teksts ar izmēru līdz 255 rakstzīmēm
- 1.8 Otrā ielas līnija – neobligāts teksts ar izmēru līdz 255 rakstzīmēm
- 1.9 Pasta kods – obligāts teksts ar izmēru līdz 255 rakstzīmēm

2 Atribūts

- 2.1 Unikāls kods – obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm
- 2.2 Vai ir ieslēgts – neobligāta Bula vērtība
- 2.3 Nosaukums – obligāts, teksts ar izmēru līdz 255 rakstzīmēm
- 2.4 Datu tips – obligāts teksts, ar 5 iespējamajiem variantiem
- 2.5 Iespējamie vērtību varianti – obligāts skaitītājs ar 5 iespējamajiem variantiem (“boolean”, “number”, “string”, “select_number”, “select_string”)
- 2.6 Vai tiek izmantots, kā filtrs meklējot produktu – neobligāta Bula vērtība

3 Atribūtu sets

3.1 Unikāls kods - obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

3.2 Vai ir ieslēgts – neobligāta Bula vertība

3.3 Nosaukums - obligāts teksts ar izmēru līdz 255 rakstzīmēm

4 Kategorija

4.1 Unikāls lapas identifikators - obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

4.2 Vai ir ieslēgta - neobligāta Bula vertība

4.3 Nosaukums - obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

4.4 Vai tiek pievienota pie “menu” - neobligāta Bula vertība

5 Pircējs

5.1 Unikāls epasts - obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

5.2 Šifrēta parole - obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

5.3 Vārds - obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

5.4 Uzvārds - obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

6 Produkts

6.1 Unikāls lapas identifikators - obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

6.2 Krājumu glabāšanas vienība – obligāts, unikāls teksts ar izmēru līdz 255 rakstzīmēm

6.3 Vai ir ieslēgts - neobligāta Bula vertība

6.4 Pamata cena -

6.5 Daudzums noliktavā – vesels skaitlis ar nenoteiktu diapazonu

6.6 Atlaides tips (daudzums, procentuāls) – neobligāts skaitītājs ar 2 iespējamām vērtībām (“percentage”, “amount”)

6.7 Atlaides vērtība - skaitlis ar diapazonu no 0 līdz bezgalības

6.8 Nodokļu procents – skaitlis ar diapazonu no 0 līdz bezgalības

6.9 Īss apraksts - obligāts, teksts ar izmēru līdz 255 rakstzīmēm

6.10 Garš apraksts – obligāts teksts ar lielu rakstzīmju daudzumu

6.11 Attēli un video – obligāta JSON tipa vērtība, kuras lauki sastāv no attēla tipa un attēla adreses. Ir 4 attēlu tipi:

- “base” – galvenais attēls
- “small” – mazs attēls
- “thumbnail” – mazākais attēls
- “other” – saraksts ar visiem pārējiem attēliem

6.12 Atribūtu vērtības – obligāta JSON tipa vērtība, kuras lauki sastāv no atribūtu koda un tā vērtības

7 Atsauksme

7.1 Status (pieņemta, apstrādā, noliegta) – skaitītājs ar 3 iespējamām vērtībām (“accepted”, “pending”, “rejected”)

7.2 Nosaukums - obligāts, teksts ar izmēru līdz 255 rakstzīmēm

7.3 Saturs - obligāts, teksts ar neierobežotu izmēru

7.4 Reitings no 1 līdz 10 – obligāts skaitlis ar diapazonu no 1 līdz 10

2.1.2 Izejas informācijas apraksts

Meklēšanas rezultātu atspoguļojums

Rēķinu izveide

2.2 Funkcionālās

1 Jānodrošina lietotāja autorizācija.

1.1 Ja ... tad ...

1.2 adsda

2 Jānodrošina pircēju informācijas apstrāde.

2.1 Adsads

2.2 adsads

2.3 Jānodrošina sdfiog

2.4 Nefunkcionālās

Dizaina apraksts

3 UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS

1 Frontend:

1.1 JavaScript – ir praktiski vienīga programmēšanas valoda, kas tiek izmantota šajā projektā, un visas darbības ar HTML, SCSS, PostgreSQL tiek veiktas arī ar JavaScript. Izvēles pamatojums ir tāds, kā šī programmēšanas valoda ļoti labi der mājaslapu veidošanai, tagad, ne tikai, lai nodrošinātu lietotāja interfeisu, bet arī, lai nodrošinātu datu apstrādi un datu bāzes manipulācijas. JavaScript ir ļoti populāra un attīstīta programmēšanas valoda, ar lielu daudzumu palīg resursu, tāpēc, izvēle ir ļoti izdevīga, jo atvieglo palīdz paātrināt izstrādes procesu.

1.2 React – interfeisa komponentu sistēma, lai labu interfeisu, ir nepieciešama laba komponentu sistēma, protams, varēja būvēt savu komponentu sistēmu, bet tas aizņemt pārkāp daudz laika, un resursu, pietam, vēl, nevar paredzēt, kādas grūtības rastos, kas tieši būtu nepieciešams, lai sistēma funkcionētu atbilstoši vajadzībām, tāpēc, ir labāk izmantot jau gatavu, pārbaudītu, sistēmu.

1.3 Material UI – bāzes stils, būtu labi, lai jau uzreiz būtu kaut kāds mājaslapas stils, kuru pēc tam pēc konkrēta lietotāja nepieciešamībām varētu modificēt un pielāgot tā specifiskām vajadzībām.

1.4 React Admin – administratora daļas abstrakcija, nodrošina resursu sasaisti starp Backend un Frontend, konkrēti administrātora daļai, izvēle, ir pamatota, ar to, ka ir nepieciešama, kaut kāda bāze, uz kuras būvēt administrātoru daļu, jo lielāka daļa no operācijām atkārtojas, tāpēc arī šīs komponents ir ļoti lietderīgs.

2 Backend:

2.1 Node – JavaScript bāzēta Backend sistēma, ar to palīdzību tiek nodrošināta visas servera operācijas: GraphQL API un PostgreSQL.

Alternatīvas izvēles varētu būt PHP, bet šajā gadījumā, lai ērtāk saskaņotu Backend ar frontend, tieši Node der ļoti labi, neskatoties uz tā trūkumiem.

2.2 Express -

2.3 PostgreSQL – datu bāzes vadības sistēma, lai nodrošinātu datu uzglabāšanu un manipulācijas. Tieši izvēlējos PostgreSQL un ne MySQL, vai citu SQL tipa datu bāzes vadības sistēmu, tāpēc, ka PostgreSQL satur specifisku funkcionalitāti, kura noderētu projekta realizācijas, konkrēti, JSONB datu tips, ar kura palīdzību var realizēt efektīvu produktu atribūtu sistēmu, protams arī izmantojot MySQL, varētu to iegūt ar EAV paņiemieniem.

2.4 GraphQL – pieprasījumu valoda, kura nodrošina efektīvu datu pieprasījumu rezolūciju. Klients pieprasa konkrētus datus, un serveris tieši tos tadus, kurus klients ir pierasījis, nevairāk, nemazāk. Atšķībā no alternatīvas – Rest API, šī sistēma ir krietni elastīgāka un datu pieprasījums tiek veikts no viena endpoint.

4 PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

4.1 Sistēmas arhitektūra

Manā sistēmā ir divi lietotāju tipi: administrators un pircējs, tāpēc kopumā interneta veikala sistēma sastāvēs no 2 daļām: administratora daļa un klienta daļa. (skatīt 3. pielikumā).

4.1.1 *Administratora daļa.*

Administrators var apskatīt un rediģēt interneta veikala datus. Tam ir pieeja pie produktiem, kategorijām, atribūtiem, atribūtu kopām, CMS lapām, CMS blokiem, ierobežotiem pircēja datiem (kartes dati un paroles tiek šifrētas), pasūtījumiem, rēķiniem, produktu atsauksmēm un citiem datiem.

4.1.2 *Klienta daļa.*

Klienta nevar redzēt un manipulēt interneta veikala datus, tas redz tikai tā priekšpusi, kurā tas spēj atrast sev nepieciešamos produktus un pievienot tos savā grozā. Lietotājs var būt kā viesis, vai arī kā reģistrēts lietotājs, abos gadījumos ir iespējams veikt pirkumu. Abu tipu lietotāji var meklēt produktu ievadot kaut kādus datus par to, piemēram, produkta nosaukumu,

vai arī izvēlēties noteiktus filtrus, un balstoties uz tiem, izlasīt noteiktus produktus. Var pievienot produktu pie groza, rediģēt tā daudzumu, vai arī noņemt to. Lietotājs var pierēģistrēties. Reģistrējoties, lietotājs var saglabāt savas adreses, kartes datus, kā arī pievienot, rediģēt un dzēst produktu atsauksmes.

4.2 Sistēmas ER modelis

ER diagramma (skatīt 1. pielikumā) sastāv no 12 entitijām, kas atspoguļo datu apriti sistēmā.

- “Product” – produkts,
- “AttributeSet” – atribūtu sets ir sets, kas apvieno kaut kādu atribūtu daudzumu.
- “Attribute” – atribūts, kas apraksta konkrētas produkta īpašības
- “Category” – kategorija, ir grupa, kas apvieno vairākas citas kategorijas (apakškategorijas), vai/un produktus.
- “Customer” – pircējs – persona, kura meklē produktus, liek tos grozā un veic pasūtījumums.
- “Address” – adrese, kuru var pievienot pircējs pirkuma laikā, un kas ir nepieciešama pasūtījumam, lai nodrošinātu piegādi un apmaksu
- “Cart” – grozs, struktūra, kas satur sevī kopsavilkumu (cenās) par tā saturu, un sastāv tas no groza priekšmetiem.
- “CartItem” – groza priekšmets, ir produkts, kas tika pievienots grozām, ar papildus datiem - daudzums.
- “Order” – pasūtījums ir pircēja darbības galīgais rezultāts, tas ietver sevī visu nepieciešamo informāciju: kopējie dati: numurs, datums, cenās: cena par piegāi, nodokļi, kopējā cena bez nodokļiem, kopējā cena ar nodokļiem, un informāciju par visiem pasūtītiem produktiem un to daudzumu.
- “OrderItem” – pasūtījuma priekšmets, kas pieder noteiktam pasūtījumam, kas saglabā sevī uz to brīdi aktuālo informāciju par produktu
- Invoice - rēķins, ir dokuments, kas satur visu galīgo informāciju par pasūtījuma datiem: kopējie dati: numurs, datums, cenās: cena par piegāi,

nodokļi, kopējā cena bez nodokļiem, kopējā cena ar nodokļiem, un referenci uz konkrētu pasūtījumu un tā pasūtījuma priekšmetiem.

- Review – atsauksme, kuru uzraksta pircējs par produktu.

Datu bāzes relācijas parada kā savstarpēji ir savienotas divas vai vairākas entītijas:

- Starp produktiem un kategorijām ir attiecība daudzi pret daudziem, jo vienam produktam var būt vairākas kategorijas, un vienai kategorijai var būt vairāki produkti.
- Starp produktiem un atribūtu setiem ir attiecība daudzi pret vienu, jo produktam var būt tikai viens atribūtu sets, bet atribūtu setiem var būt vairāki produkti.
- Starp produktu un atsauksmi ir attiecība viens pret daudziem, jo vienam produktam var būt vairākas atsauksmes, bet vienai atsaukmei var būt tikai viens produkts.
- Starp pircēju un adresi ir attiecība viens pret daudziem, jo vienam pircējam var būt vairākas adreses, bet vienai adresei var būt tikai viens pircējs.
- Starp pircēju un produktu ir attiecība daudzi pret daudziem, jo viens pircējs var nopirkt vairākus produktus, un viens produkts var būt pārdots vairākiem pircējiem.
- Starp pircēju un pasūtījumu ir attiecība viens pret daudziem, jo vienam pircējam var būt vairāki pasūtījumi, bet pasūtījumam var būt tikai viens pircējs.
- Starp pircēju un atsauksmi ir attiecība viens pret daudziem, jo vienam pircējam var būt vairākas atsauksmes, bet vienai atsaukmei var būt tikai viens pircējs.
- Starp pircēju un pircēja grupu ir attiecība daudzi pret daudziem, jo vienam pircējam var būt vairākas pircēju grupas, un vienai pircēju grupai var būt vairāki pircēji.
- Starp kategoriju un kategoriju ir attiecība daudzi viens pret daudziem, jo vienai kategorijai var būt vairākas apakškategorijas, bet vienai apakškategorijai, var būt tikai viena kategorija.
- Starp pasūtījumu un rēķinu ir attiecība viens pret viens, jo vienam pasūtījumam var būt tikai viens rēķins un vienam rēķinam var būt tikai viens pasūtījums.

- Starp atribūtu setu un atribūtu ir attiecība daudzi pret daudziem, jo vienam atribūtu setam var būt vairāki atribūti, un vienam atribūtam var būt vairāki atribūtu seti.

4.3 Funkcionālais sistēmas modelis

5 DATU STRUKTŪRAS APRAKSTS

Datu bāzes fiziskā struktūra (skatīt 2. pielikumā) sastāv no 16 tabulām, kas satur informāciju par sistēmas lietotājiem, klientiem, kategorijām, produktiem, pirkumiem, atribūtiem un citiem objektiem. Ir divas galvenās tabulas, kurām ir visvairāk savienojumu, tas ir produktu tabula, un klientu tabula, un, acīmredzami, šie divi objekti, kopumā arī definē lielāko daļu no sistēmas, jo klients pērk produktus. Datubāzes realizēšanai izmanto PostgreSQL. Shēma tika ģenerēta.

Tabulā “Address” glabājas informācija par klienta izveidotajām adresēm.

Šī tabula attiecās uz “Customer” tabulu ar “customer_id” ārējo atslēgu, un viedo “daudzi pret vienu” attiecību – vairākas adreses var piederēt vienam klientam.

1. tabula

Tabulas “Address” struktūra

Numurs	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	firstName	varchar	255	Vārds
3.	lastName	varchar	255	Uzvārds
4.	phoneNumber	varchar	255	Telefona numurs
5.	country	varchar	255	Valsts
6.	city	varchar	255	Pilsēta
7.	province	varchar	255	Reģions
8.	street1	varchar	255	Iela
9.	street2	varchar	255	Iela
10.	postalCode	varchar	255	Pasta kods
11.	customer_id	integer	-	Ārēja atslēga uz klientu

Tabulā “Attribute” glabājas informācija par produkta atribūtiem.

Šī tabula attiecās uz “AttributeSet” tabulu un viedo “daudzi pret daudziem” attiecību – vairāki atribūti var piederēt vairākiem atribūtu setiem, tāpēc ir nepieciešama atsevišķa tabula “AttributeSetAttribute”, kur ir divas ārēja atslēgas, kas savieno šīs tabulas: “attribute_set_id”, “attribute_id”.

Tabulas “Attribute” struktūra

Numurs	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	code	varchar	255	Unikāls kods
3.	isEnabled	boolean	-	Vai ir ieslēgts
4.	label	varchar	-	Nosaukums
5.	type	varchar	255	Datu tips
6.	attributeOptions	jsonb	-	Iespējamie vērtību varianti
7.	isFilter	boolean	-	Vai tiek izmantots, kā filtrs meklējot produktu

Tabulā “AttributeSet” glabājas informācija par atribūtu setiem, tās ir kopas, kas apvieno vairākus atribūtus, un produktam var būt viena atribūtu kopa, un uz to kopu bāzēsies tā iespējamie atribūti.

Šī tabula attiecās uz “Attribute” tabulu un viedo “daudzi pret daudziem” attiecību – vairāki atribūti var piederēt vairākiem atribūtu setiem, tāpēc ir nepieciešama atsevišķā tabula “AttributeSetAttribute”, kur ir divas ārēja atslēgas, kas savieno šīs tabulas: “attribute_set_id”, “attribute_id”.

Tabulas “AttributeSet” struktūra

Numurs	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	code	varchar	255	Unikāls kods
3.	isEnabled	boolean	-	Vai ir ieslēgts
4.	name	varchar	255	Nosaukums

Tabulā “AttributeSetAttribute” ir tabula, lai realizēta “daudzi pret daudziem” attiecību starp “AttributeSet” un “Attribute” tabulām.

Tabulas “AttributeSetAttribute” struktūra

Numurs	Nosaukums	Tips	Garums	Piezīme
1.	attribute_set_id	integer	-	Ārēja atslēga uz atribūtu setu

2.	attribute_id	integer	-	Ārēja atslēga uz atribūtu
----	--------------	---------	---	---------------------------

Tabulā “Cart” (nepieciešams papildināt tabulas laukus) glabājas informācija par klienta grozu.

Šī tabula attiecās uz “Customer” tabulu ar “customer_id” ārējo atslēgu, un veido “viens pret vienu”, jo vienam klientam var būt tikai viens grozs.

5. tabula

Tabulas “Cart” struktūra

Numurs	Nosaukums	Tips	Garums	Piezīme
1.	totalTax	double	-	
2.	subtotal	double	-	
3.	total	double	-	
4.	customer_id	integer	-	Ārēja atslēga uz klientu

Tabulā “CartItem” glabājas informācija par katru pievienotu produktu pie klienta groza.

Šī tabula ir nepieciešama, lai realizētu “daudzi pret daudziem” attiecību starp “Cart” un “Product” tabulām.

6. tabula

Tabulas “CartItem” struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	quantity	integer	-	Produkta daudzums
2.	cart_id	integer	-	Ārēja atslēga uz grozu
3.	product_id	integer	-	Ārēja atslēga uz produktu

Tabulā “Category” glabājas informācija par kategorijām.

7. tabula

Tabulas “Category” struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga

2.	urlKey	varchar	255	Unikāls lapas identifikators
3.	isEnabled	boolean	-	Vai ir ieslēgta
4.	name	varchar	255	Nosaukums
5.	isInMenu	boolean	-	Vai tiek pievienota pie "menu"
6.	category_id	integer	-	Ārēja atslēga uz pašu sevi

Tabulā "Customer" glabājas informācija par klientiem.

8. tabula

Tabulas "Customer" struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	email	varchar	255	Unikāls epasts
3.	password	varchar	255	Šifrēta parole
4.	firstName	varchar	255	Vārds
5.	lastName	varchar	255	Uzvārds
6.	token	text	-	Autorizācijas tokens
7.	isGuest	boolean	-	Vai ir viesis

Tabulā "CustomerCustomerGroup" glabājas informācija par klientu grupu grupām.

Šī tabula ir nepieciešama, lai realizētu "daudzi pret daudziem" attiecību starp "Customer" un "CustomerGroup" tabulām.

9. tabula

Tabulas "CustomerCustomerGroup" struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	customer_id	integer	-	Ārēja atslēga uz klientu
2.	customer_group_id	integer	-	Ārēja atslēga uz klienta grupu

Tabulā "CustomerGroup" glabājas informācija par klientu grupām.

10. tabula

Tabulas "CustomerGroup" struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
-----	-----------	------	--------	---------

1.	customer_id	integer	-	Ārēja atslēga uz klientu
2.	customer_group_id	integer	-	Ārēja atslēga uz klienta grupu

Tabulā “Invoice” glabājas informācija par klienta rēķiniem.

11. tabula

Tabulas “Invoice” struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	date	date	-	Datums
3.	totalDelivery	double	-	Pārvešanas izmaksas
4.	totalTax	double	-	Nodokļi
5.	subtotal	double	-	Kopējā cena bez nodokļiem
6.	total	double	-	Kopējā cena
7.	order_id	integer	-	Ārēja atslēga uz pirkumu

Tabulā “Order” glabājas informācija par pirkumiem.

12. tabula

Tabulas “Order” struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	reference	varchar	255	Unikāla eference
3.	date	date	-	Datums
4.	status	varchar	255	Status
5.	totalDelivery	double	-	Pārvešanas izmaksas
6.	totalTax	double	-	Nodokļi
7.	subtotal	double	-	Kopēja cena bez nodokļiem
8.	total	double	-	Kopēja cena
9.	customer_id	integer	-	Ārēja atslēga uz klientu
10.	address_id	integer	-	Ārēja atslēga uz adresi

Tabulā “OrderItem” glabājas informācija par pirkumiem.

Tabulas “OrderItem” struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	quantity	integer	-	Produkta daudzums
3.	totalTax	double	-	Nodokļi
4.	subtotal	double	-	Kopēja cena bez nodokļiem
5.	total	double	-	Kopēja cena
6.	order_id	integer	-	Ārēja atslēga uz pirkumu
7.	product_id	integer	-	Ārēja atslēga uz produktu

Tabulā “Product” glabājas informācija par produktiem.

Tabulas “Product” struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	urlKey	varchar	255	Unikāls lapas identifikators
3.	sku	varchar	255	Krājumu glabāšanas vienība
4.	isEnabled	boolean	-	Vai ir ieslēgts
5.	price	double	-	Pamata cena
6.	stockQuantity	integer	-	Daudzums noliktavā
7.	specialDiscountType	varchar	255	Atlaides tips (daudzums, procentuāls)
8.	specialDiscountValue	double	-	Atlaides vērtība
9.	specialTaxRate	double	-	Nodokļu procents
10.	shortDescription	varchar	255	Īss apraksts
11.	longDescription	text	-	Garš apraksts
12.	media	jsonb	-	Attēli un video
13.	attributeValues	jsonb	-	Atribūtu vērtības
14.	attribute_set_id	integer	-	Ārēja atslēga uz atribūtu

Tabulā “ProductCategory” glabājas informācija par produktu kategorijām.

Šī tabula ir nepieciešama, lai realizētu “daudzi pret daudziem” attiecību starp “Product” un “Category” tabulām.

15. tabula

Tabulas “ProductCategory” struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	product_id	integer	-	Ārēja atslēga uz produktu
2.	category_id	integer	-	Ārēja atslēga uz kategoriju

Tabulā “Review” glabājas informācija par klienta atsauksmēm uz produktu.

16. tabula

Tabulas “Review” struktūra

Nr.	Nosaukums	Tips	Garums	Piezīme
1.	id	integer	-	Primāra atslēga
2.	status	varchar	255	Status (pieņemta, apstrādā, noliegta)
3.	date	date	-	Datums
4.	title	varchar	255	Nosaukums

5.	content	text	-	Saturs
6.	rating	integer	-	Reitings no 1 līdz 10
7.	customer_id	integer	-	Ārēja atslēga uz klientu
8.	product_id	integer	-	Ārēja atslēga uz produktu

6 LIETOTĀJA CEĻVEDI

6.1 Sistēmas prasības aparatūrai un programmatūrai

Nav specifiskas sistēmas aparatūras prasības, un programmu ir iespējams palaist gandrīz uz jebkuras modernās sistēmas, tāpēc, visas prasības attiecās tikai uz nepieciešamo programmatūru. Programma var būt instalēta gan uz Linux, gan uz Windows, taču MacOS netiek atbalstīts šobrīd, lai instalētu un palaistu to ir nepieciešama noteikta programmatūra:

- Node / NPM
- PostgreSQL

6.1.1 Windows

Lai šo programmu varētu palaist ir nepieciešams uzinstalēt

1 Node / NPM

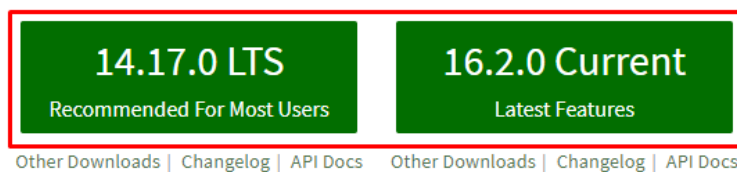
1.1 Pāriet uz šo adresi: <https://nodejs.org/en/>

1.2 Tad lejupielādēt 14.17.0 LTS versiju, taču programma spēj strādāt arī ar visjaunāko Node versiju

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Join us at OpenJS World, a free virtual event on June 2-3, 2021

Download for Windows (x64)



Or have a look at the [Long Term Support \(LTS\) schedule](#).

1. att. Node / NPM instalācija

1.3 Palaist instalāciju un izpildīt visas darbības pēc noklusējuma

1.4 Pēc šo darbību izpildes, programma būs veiksmīgi instalēta

2 PostgreSQL

- 2.1 Pāriet uz šo adresi: <https://www.postgresql.org/download/windows/>
- 2.2 Tad lejupielādē jaunāko PostgreSQL versiju

Windows installers

Interactive installer by EDB

Download the installer certified by EDB for all supported PostgreSQL versions.

This installer includes the PostgreSQL server, pgAdmin; a graphical tool for managing and developing your PostgreSQL tools and drivers. Stackbuilder includes management, integration, migration, replication, geo:

This installer can run in graphical or silent install modes.

The installer is designed to be a straightforward, fast way to get up and running with PostgreSQL on Windows.

Advanced users can also download a **zip archive** of the binaries, without the installer. This download is in

2. att. PostgreSQL instalācija

- 2.3 Tad izpildīt visas darbības pēc noklusējuma.
- 2.4 Pēc šo darbību izpildes, programma būs veiksmīgi instalēta.

6.1.2 Linux (Ubuntu)

Lai uzinstalētu visu uz Linux (ubuntu), ir nepieciešams palaist tikai dažas komandas:

1 Node / NPM

- 1.1 `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash`

- 1.2 `nvm install 14.17.0`

- 1.3 Pēc šo darbību izpildes, programma būs veiksmīgi instalēta

2 PostgreSQL

- 2.1 `sudo apt update`
- 2.2 `sudo apt install postgresql postgresql-contrib`
- 2.3 Pēc šo darbību izpildes, programma būs veiksmīgi instalēta

6.2 Sistēmas instalācija un palaišana

Vispirms ir nepieciešams strādājošs Node / NPM un PostgreSQL setups, kad tas ir sagatavots, tad ir nepieciešams palaist sekojošas komandas:

1. git clone <https://github.com/rvtprog-kval-21/d41-DenissStrombergs-InternetaVeikals.git>
2. cd d41-DenissStrombergs-InternetaVeikals
3. cd backend
4. npm start
5. Atvert jaunu konsoles logu DenissStrombergs-InternetaVeikals direktorijā
6. cd frontend
7. npm start

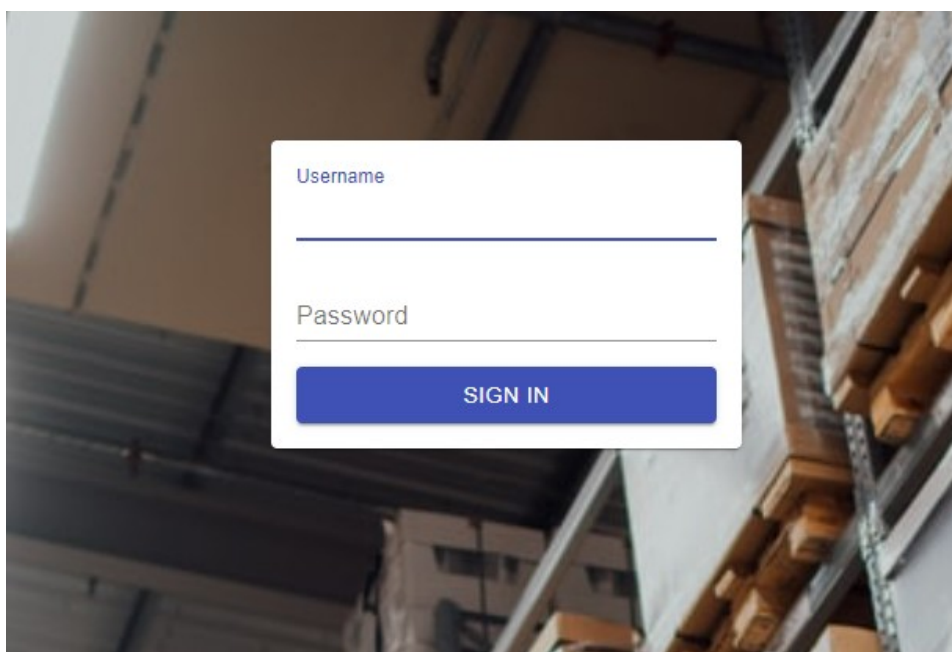
6.3 Programmas apraksts

6.3.1 *Administrators*

Administratora daļas apraksts:

1 Autorizācija

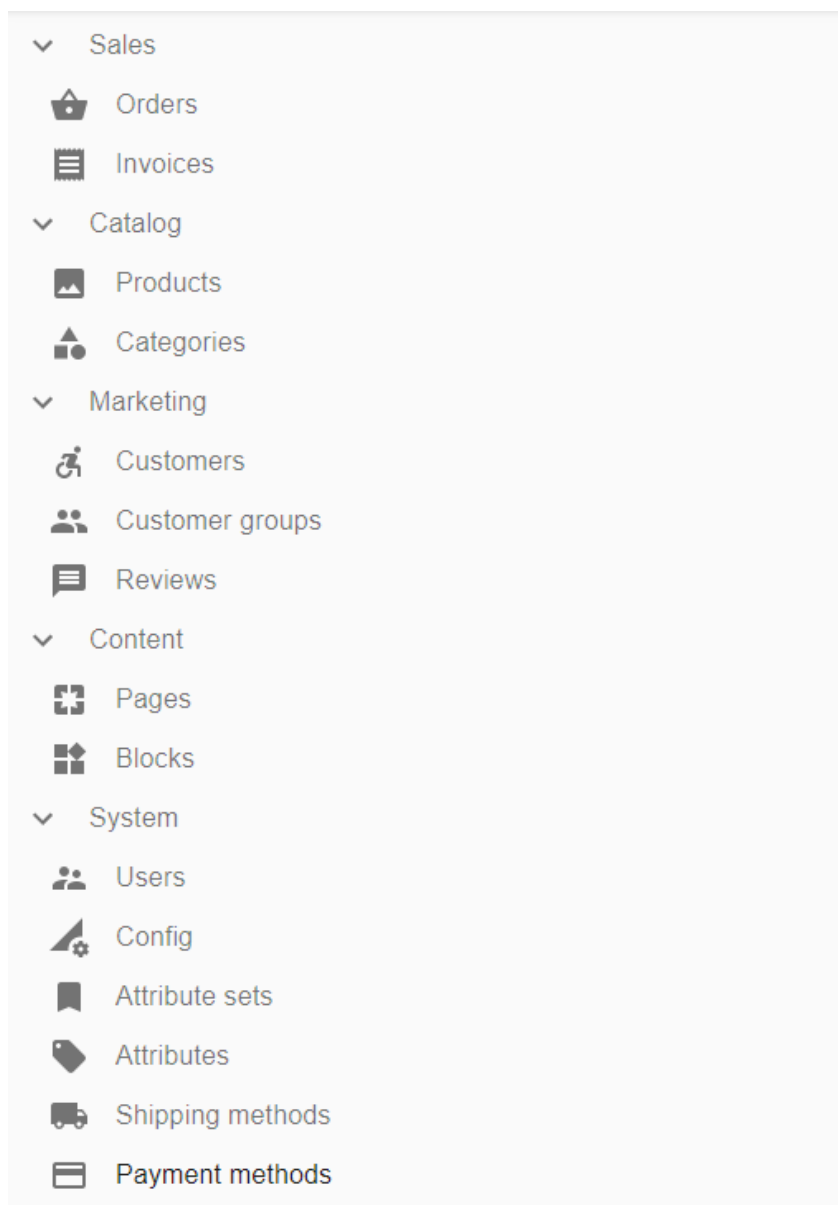
- 1.1 Lai autorizēties, kā administrators, ir nepieciešams pāriet uz “/admin” lapu



3. att. Autorizācijas forma

- 1.2 Tad ievadīt lietotāja autorizācijas datus, pēc noklusējuma, lietotāj vārds ir **admin** un parole ir arī **Admin1234**

- 1.3 Pēc lietotāja datu ievades, ir nepieciešams uzspiest uz “Sign in” pogu, un tad, ja lietotājvārds un parole tika veiksmīgi ievadītas, tad jūs pāriesiet uz administratora lapu
- 1.4 Pēc autorizācijas var redzēt navigācijas paneli, tajā ir iespējams redzēt, visus administratora pārvaldes sadalījumus, spiežot uz konkrētu sadalījumu lietotājs var pāriet uz noteiktu lapu, kur būs redzami visi atbilstoši dati



4. att. Navigācijas panelis

2 Lietotāju pārvalde

2.1 Lai redzētu, kuriem lietotājiem ir piekļuve pie administratora daļas ir nepieciešams pāriet uz “User” sadalījumu, šajā vietā var arī uzspiest pogu “Create”, lai pievienotu jaunu lietotāju, vai “Edit”, lai apskatītu vai rediģētu jau eksistējošu lietotāju:

[+ CREATE](#)

<input type="checkbox"/>	Id ↑	Username	Email	First name	Last name	
<input type="checkbox"/>	1	deniss	deniss.strombergs@scandiweb.com	Deniss	Strombergs	EDIT
<input type="checkbox"/>	2	nikolajs	nikolajs.lapins@gmail.com	Nikolajs	Lapins	EDIT
<input type="checkbox"/>	3	anna	anna.kaulina@gmail.com	Anna	Kaulina	EDIT
<input type="checkbox"/>	4	daniela	daniela.salmina@gmail.com	Daniela	Salmina	EDIT


Rows per page: 10 ▾ 1-4 of 4

5. att. Lietotāju saraksts

2.2 Pārejot vai nu uz “Create” vai uz “Edit”, parādīsies lauki, kuros vajadzēs ierakstīt nepieciešamus datus par lietotāju, lai to varētu saglabāt.


Username

Email

Password 

First name

Last name

 SAVE

6. att. Lietotāja rediģēšanas lapa

3 Konfigurācijas pārvalde

3.1 Lai redzētu kopējo veikala konfigurāciju, ir nepieciešams pāriet uz “Config” sadaļu.

3.2 Tajā var redzēt visu iestatījumus

Currency sign

€

SAVE

7. att. Konfigurācija

3.3 Lai redzētu visas piegādes veidus ir nepieciešams pāriet uz “Shipping methods” sadaļu

[+ CREATE](#)
[↓ EXPORT](#)

<input type="checkbox"/>	Id ↑	Code	Name	
<input type="checkbox"/>	1	flat_rate	Flat rate	EDIT
<input type="checkbox"/>	2	table_rate	Table rate	EDIT

Rows per page: 10 ▼

1-2 of 2

8. att. Piegādes veidu saraksts

3.4 Lai redzētu visas apmaksas veidus ir nepieciešams pāriet uz “Payment methods”

[+ CREATE](#)
[↓ EXPORT](#)

<input type="checkbox"/>	Id ↑	Code	Name	
<input type="checkbox"/>	1	bank_transfer	Bank transfer	EDIT
<input type="checkbox"/>	2	payment_card	Payment card	EDIT

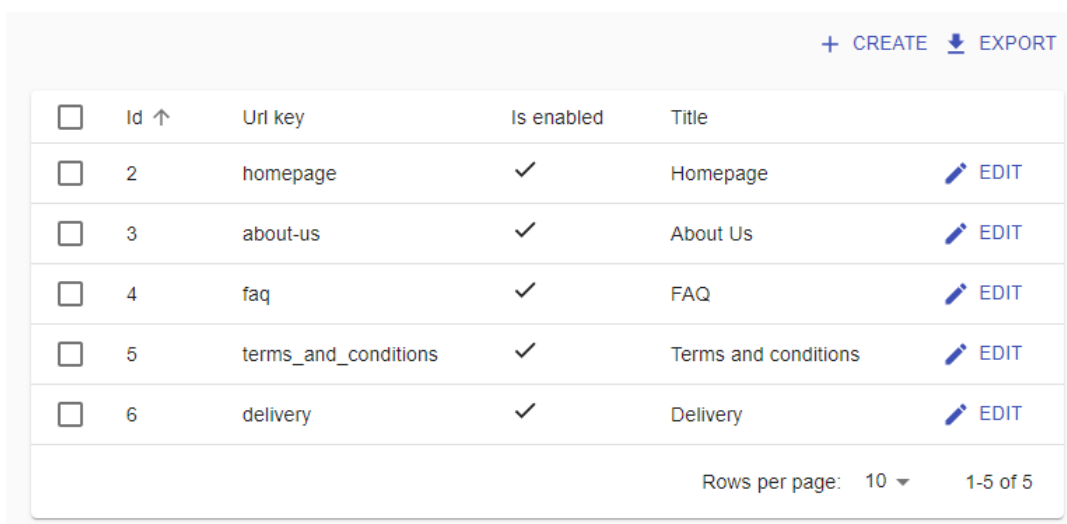
Rows per page: 10 ▼

1-2 of 2

9. att. Apmaksas veidu saraksts

- Satura pārvalde, tā ir “Content” sadaļa, kas satur 2 apakšsadaļas, “Page”, lai redzētu un rediģētu visas specifiskas lapas un “Block”, lai redzētu un rediģētu visas atsevišķas lapas komponentus.

4.1 Lai redzētu visas lapas ir nepieciešams pāriet uz “Pages”, šajā vietā var arī uzspiest pogu “Create”, lai pievienotu jaunu lapu, vai “Edit”, lai apskatītu vai rediģētu jau eksistējošu lapu:



<input type="checkbox"/>	Id ↑	Url key	Is enabled	Title	
<input type="checkbox"/>	2	homepage	✓	Homepage	EDIT
<input type="checkbox"/>	3	about-us	✓	About Us	EDIT
<input type="checkbox"/>	4	faq	✓	FAQ	EDIT
<input type="checkbox"/>	5	terms_and_conditions	✓	Terms and conditions	EDIT
<input type="checkbox"/>	6	delivery	✓	Delivery	EDIT
Rows per page: 10 1-5 of 5					

10. att. Lapu saraksts

4.2 Pārejot vai nu uz “Create” vai uz “Edit”, parādīsies lauki, kuros vajadzēs ierakstīt nepieciešamus datus par lietotāju, lai to varētu saglabāt.

- 5 Produktu pārvalde
- 6 Kategoriju pārvalde
- 7 Atribūtu setu pārvalde
- 8 Atribūtu pārvalde
- 9 Klientu pārvalde
- 10 Klientu grupu pārvalde
- 11 Pasūtījumu pārskats
- 12 Rēķinu pārskats
- 13 Atsauksmju pārskats

6.3.2 *Klients*

Klienta daļas apraksts:

- 1 Mājaslapa un citas lapas
- 2 Produktu meklēšana

- 3 Produktu apskatīšana
- 4 Groza rediģēšana
- 5 Pirkuma veikšana
- 6 Lietotāja reģistrācija un autorizācija
- 7 Adrešu pārvalde
- 8 Pasūtījumu pārvalde
- 9 Atsauksmju pārvalde
- 10 Atsauksmju pievienošana

6.4 Testa piemērs

Detalizēta produktu pievienošanas instrukcija.

NOBEIGUMS

Kāda struktūra?

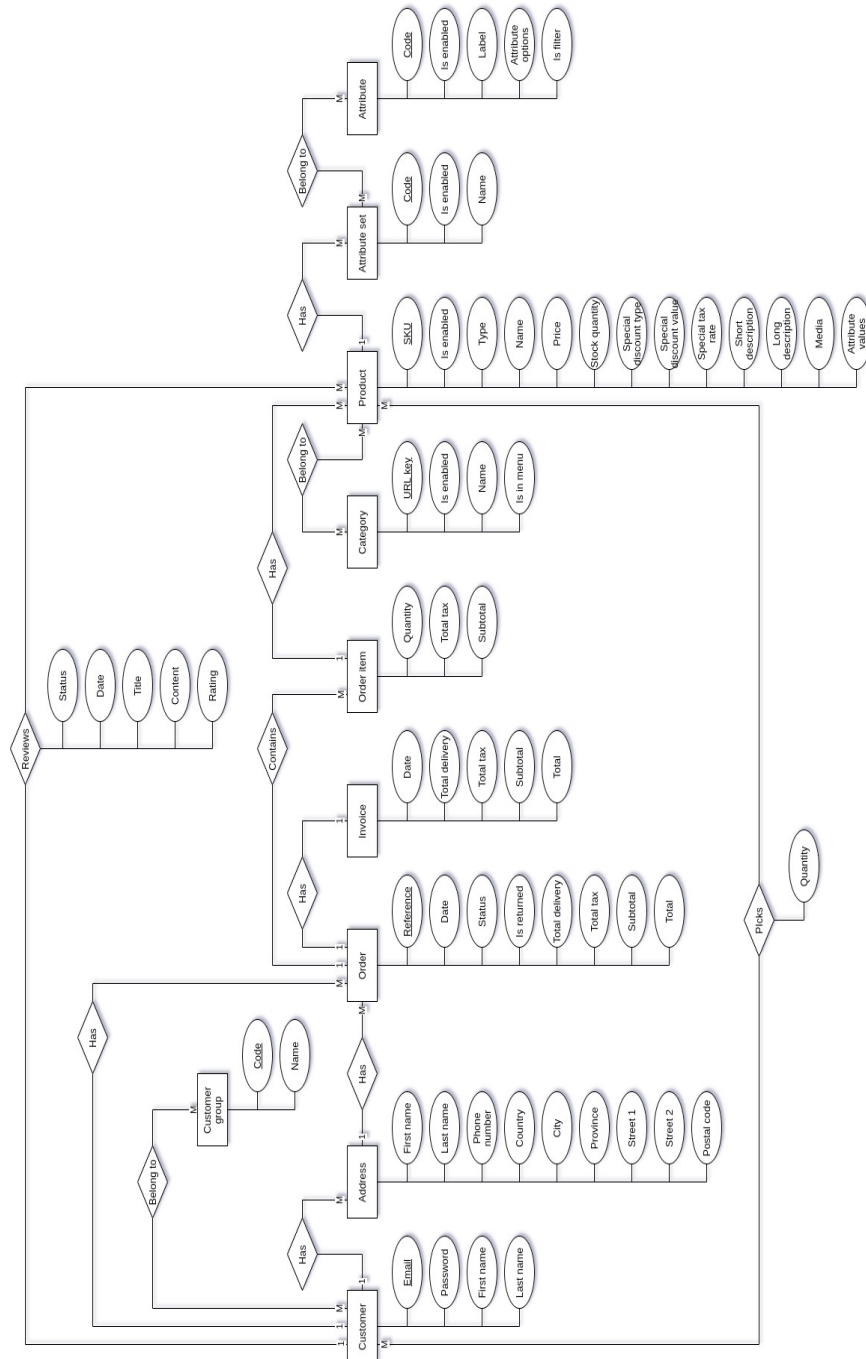
INFORMĀCIJAS AVOTI

1. <https://marmelab.com/react-admin/Readme.html>

PIELIKUMI

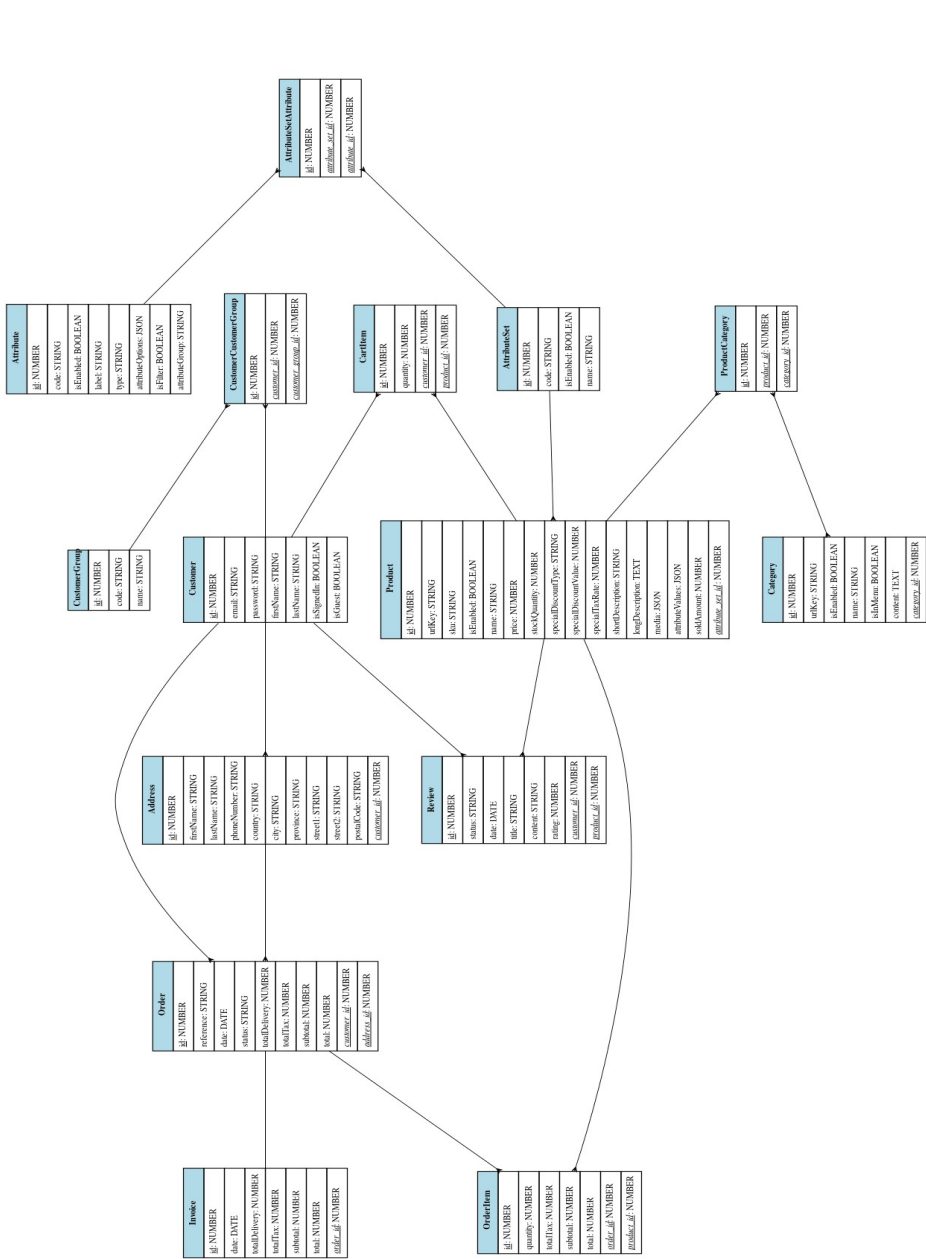
1. pielikums

ER diagramma



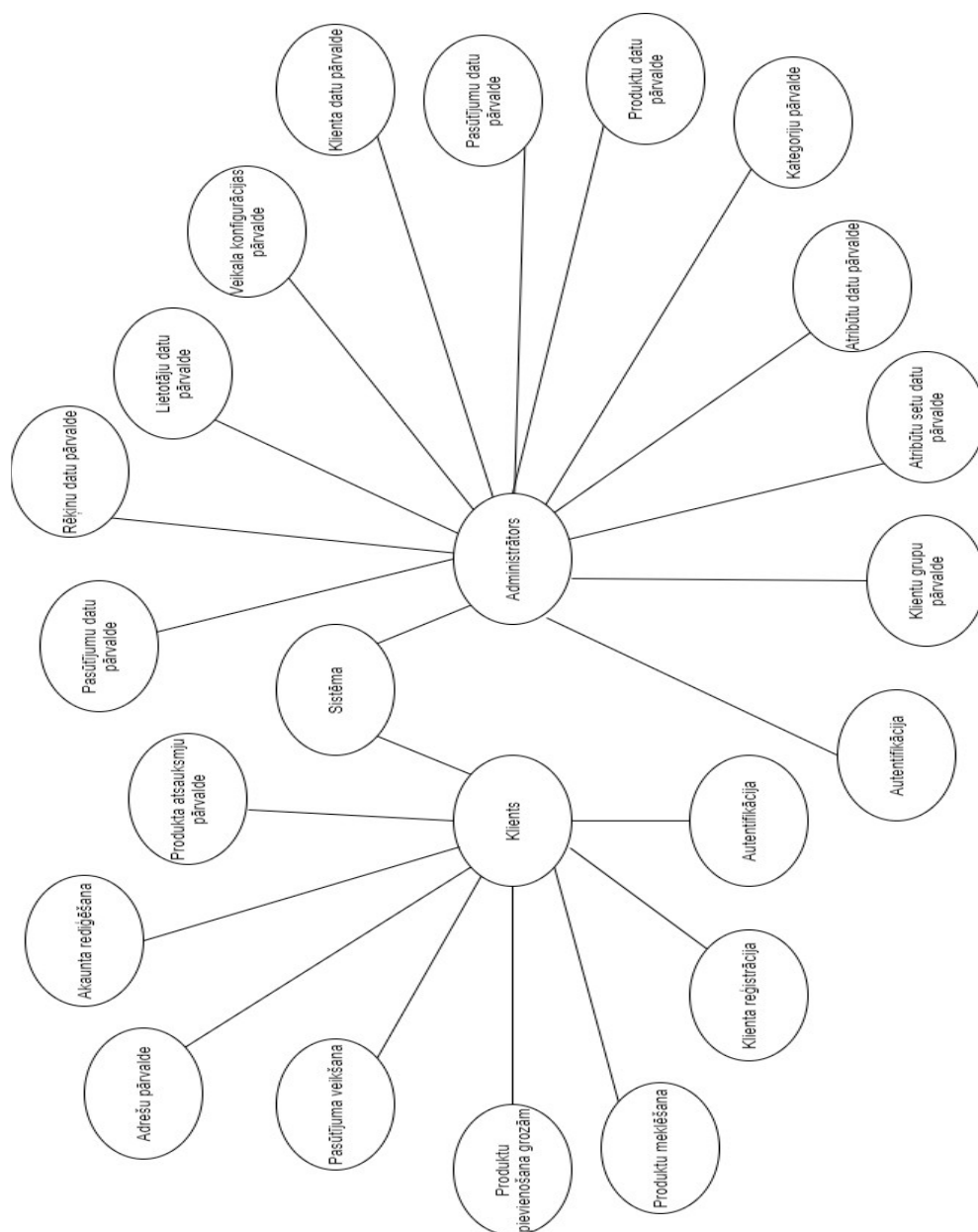
1. attēls. ER diagramma

Fiziska struktūra



2. attēls. Fiziska struktūra

Sistēmas arhitektūra



3. attēls. Sistēmas arhitektūra