



Liepājas Valsts tehnikums

Preču mazumtirdzniecības e-veikals “Piffdeals”

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija

33484011 Programmēšana

Grupas nosaukums

Programmēšanas tehniķis

Projekta izstrādātājs **Markuss Balodis**

/vārds, uzvārds, paraksts/

Eksāmena datums 2024. gada 20.jūnijs

Liepāja 2024

Saturs

Ievads	4
1.Uzdevuma formulējums	5
2. Programmatūras prasību specifikācija.....	6
2.1. Produkta perspektīva.....	6
2.2. Sistēmas funkcionālās prasības	6
2.2.1. Klienta sadaļas lapa.....	6
2.2.2. Preču pasūtīšana un preces.....	11
2.2.3. Lietotāja kots.....	14
2.2.4. Administrācijas panelis	18
2.2.5. Citi.....	22
2.3. Sistēmas nefunkcionālās prasības	23
2.4. Gala lietotāja raksturozīmes	23
3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums.....	24
3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts	24
3.1.1. Back-end puses programmēšanas valodas līdzekļi	24
3.1.2. Front-end puses programmēšanas valodas līdzekļi.....	25
3.1.1. Izmantotie rīki	26
3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts.....	27
3.3. Līdzekļu un rīku lietojuma pamatojums	28
4. Sistēmas modelēšana un projektēšana.....	29
4.1. Sistēmas struktūras modelis	29
4.1.1. ER diagramma.....	29
4.2. Funkcionālais un dinamiskais sistēmas modelis	31
4.2.1. Aktivitāšu diagramma (Activity)	31
4.2.2. Lietojumgadījumu diagramma (Use Case)	32
4.3. Sistēmas moduļu apraksts un algoritmu shēmas.....	34

4.3.1. Google OAuth pieslēgšanās algoritmu shēma	34
4.3.2. Paroles atjaunošanas algoritms	36
5. Lietotāju ceļvedis	37
5.1. Reģistrācija e-veikalā kā viesis	37
5.2. Pieslēgšanās e-veikalā, lietojot Google OAuth.....	38
5.3. Konta dzēšana profilu sadaļā	38
5.4. Paroles atjaunināšana caur vēstuli.....	39
5.5. Jaunu preču izveidošana.....	42
5.6. Profila informācijas atjaunināšana	43
5.7. Paroles maiņa profila sadaļā	44
5.8. Kontakta veidlapas aizpildīšana.....	44
5.9. Preču ievietošana grozā.....	45
6. Testēšanas dokumentācija	47
6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums	47
6.2. Alternatīvās testēšanas metodes un rīki	47
6.3. Testpiemēru kopa.....	48
6.4. Testēšanas žurnāls.....	53
7. Secinājumi.....	56
8. Lietoto terminu un saīsinājumu skaidrojumi.....	57
9. Literatūras un informācijas avotu saraksts	59
Pielikums.....	60

Ievads

Kvalifikācijas eksāmena praktiskās daļas dokumentācija ir izstrādāta, lai sniegtu lasītājam skaidrību par preču mazumtirdzniecības tīmekļa vietni. Šī dokumentācija kalpo kā informatīvs resurss, lai palīdzētu izprast tīmekļa vietnes darbību, tās konfigurāciju un iespējas, kā arī nodrošināt efektīvu, un elastīgu klienta iepirkšanos tiešsaistē.

Sadaļa “Uzdevuma formulējums” sniegs lasītājam izpratni par galā sasniedzamo rezultātu sakarā ar preču mazumtirdzniecības e-veikalu. Sadaļā tiks norādīts mērķa ceļojums un e-veikala programmatūras nepieciešamību ikdienas tīmekļa vietnes lietotājam.

Sadaļa “Programmatūras prasību specifikācija” sniegs lasītājam produkta perspektīvu sakarā ar citu līdzīgu vietņu konkurentiem un norādīs to priekšrocības. Sistēmu funkcionālo un nefunkcionālo nodaļās būs iekļautas visas apkopotas prasībās sakarā ar vietnes izstrādi.

Sadaļa “Izstrādes līdzekļu, rīku apraksts un pamatojums” uzskaitīs visus līdzekļus, kuras tika izmantotas vietnes izstrādes procesā. Sadaļā tiks uzskaitīti visi līdzekļi un to rīku, kā arī vietnes programmēšanas valodu apraksts. Lasītājs vēl gūs perspektīvu par citiem alternatīviem līdzekļiem, rīkiem un citām valodām.

Sadaļa “Sistēmas modelēšana un projektēšana” sniegs lasītājam padziļinātu izpratni par vietni caur diagrammām un modeļiem. Sadaļā paredzēts iekļaut sistēmas struktūras modeli un funkcionālo, un dinamisko sistēmas modeli, piemēram, kā sekvenču, stāvokļu, aktivitāšu vai lietojumgadījumu diagrammas. Jāiekļauj sistēmas moduļu aprakstus un algoritmu shēmas, kā vissīkāk lietoto datu struktūras aprakstu un to izvēles pamatojumu.

Sadaļa “Lietotāju ceļvedis” ir izstrādāta, lai sniegtu lasītājam un vietnes lietotājam pamācību kā orientēties mājaslapā.

Sadaļa “Testēšanas dokumentācija” apraksta izstrādātāju izvēlētās testēšanas metodes, visus rīkus un to pamatojumus. Galvenais šajā sadaļā būs testēšanas žurnāls, kurā tiks apkopoti visi testpiemēri un uzskaitīti lietotie testēšanas metodes.

Sadaļas “Secinājumi”, “Saīsinājumu un terminu skaidrojumi” un “Informācijas avotu saraksts” sniegs izpratni par izstrādātāju secinājumiem un lietotajiem terminiem, kā arī avotu no kā guva informācija par vietnes tapšanu procesu.

Sadaļā “Pielikumi” saturēs 200 komandrindu garš programmētāja koda fragments, kurš būs kvalitatīvi sarakstīts un labi caurskatāms tās lasītājiem.

1.Uzdevuma formulējums

"Piffdeals.lv" preču mazumtirdzniecības e-veikals tiecas piedāvāt lietotājiem ērtu un daudzveidīgu digitālo iepirkšanās pieredzi. Lai nodrošinātu šo mērķi, izstrādātājiem ir jāpieliek pūles, lai izveidotu intuitīvu un pievilcīgu lietotāja saskarni, kas veicina gludu pāreju starp preču sortimentu, pasūtījumu veikšanas iespējām un konta pārvaldību. Tīmekļa vietnes būtiskākā sastāvdaļa varētu būt lietotāju autorizācijas sistēma, lai klientiem būtu iespēja saglabāt un apskatīt savu iepirkšanās vēsturi un vēlāk veikt vieglus, atkārtotus pasūtījumus.

Administratoriem ir jābūt pilnvarotiem un efektīviem, pasūtījumu un preču pārvaldībā. Iekļautā administratora informācijas panelis ir galvenais rīks, lai uzraudzītu vietnes darbību no abiem skatpunktiem. Tas sniedz iespēju detalizēti pārvaldīt esošo preču klāstu, pievienot jaunas preces un, kā arī uzraudzīt un optimizēt pasūtījumu izpildes procesus, nodrošinot vispusīgu kontroli pār platformu.

Ņemot vērā konkurences analīzi, izstrādātājiem ir jāorientējas uz pievienoto vērtību vietnei, lai piesaistītu klientus un izceltos konkurentu vidū caur inovatīvām funkcijām un ar uzlaboto preču katalogu, kā arī ar SEO optimizāciju, kuri var būt svarīgi faktori, lai veicinātu vietnes redzamību un atšķirtu to no citiem tirgus konkurentiem. To varētu uzskatīt par programmatūras mugurkaulu, kuras galvenais mērķis ir nodrošināt efektīvu vietnes darbību, netraucējot klientu apkalpošanu un efektīvu administratīvo pārvaldību.

Nepieciešamā programmatūra ir galvenais elements, lai nodrošinātu efektīvu vietnes darbību, klientu apkalpošanu un administratīvo pārvaldību. Šī programmatūra jāizvēlas, ņemot vērā visus projektam izvirzītos mērķus un prasības, lai nodrošinātu sistēmas stabilitāti un drošību. Izstrādātājiem ir jānodrošina, ka programmatūra ir pielāgota klientu un administratoru vajadzībām, lai ļautu tiem sasniegt savus mērķus efektīvi un veiksmīgi. Turklāt, šie elementi ne tikai saskan ar projekta mērķiem, bet arī apliecina apņēmību nodrošināt izcilu lietotāja pieredzi, nodrošinot platformas panākumus konkurētspējīgā e-komercijas vidē.

Tomēr galvenā uzmanība jāpievērš ne tikai pašas vietnes izveidei. Izstrādātājiem arī jāizpēta iespējas uzlabot lietotāja pieredzi dažādi, piemēram, ieviešot papildu funkcijas, ka lietotājus varētu informēt par jaunākajiem produktiem caur e-pasta sūtīšanu, vai saņemt informāciju par savu iecienīto preci sūtījumu statusu noliktavā. Šīs papildu funkcijas veicina visaptverošu un saistošu lietotāja pieredzi, nodrošinot platformas panākumus konkurētspējīgā e-veikala vidē. Nepieciešamai, izvēlētajai programmatūrai jāatbilst visiem projekta standartiem, uzsverot stabilitāti, drošību un pielāgošanu klientu un administratoru vajadzībām, lai efektīvi un veiksmīgi sasniegtu projekta mērķus.

2. Programmatūras prasību specifikācija

Programmatūras prasību specifikāciju nodaļas ir paredzētas, lai detalizēti aprakstīt tīmekļa vietnes funkcionālās un nefunkcionālās prasības, kā arī vietnes produktu perspektīvu un gala lietotāju raksturiezīmes. Produkta perspektīva tiek sarakstīta, lai informētu lasītāju par vietnes nākotnes plāniem un konkurentu salīdzināšanu, savukārt, gala lietotāju raksturiezīmju ietvaros ir informēt lasītāju par e-veikala nepieciešamību un tās mērķauditoriju.

2.1. Produkta perspektīva

Pēc klienta pasūtījuma, e-veikala pirmā izstrādājamā versijā tikai sastāvēs no klienta lapas un administrācijas paneļa, kurā vietnes administrators ir tiesīgs veikt CRUD funkcijas saistībā ar preču izklāsta sortimentu. Klientam interesē norādīt savu nepieciešamo kontaktinformāciju un preču pasūtījuma gadījumos, klientam būtu jāsaazinās ar piegādātāju, šai gadījumā ar manu pašu klientu. Nākotnes plānos ietilpst mājaslapas pilnveidošana ar iespēju veikt dažādu preču pasūtīšanu, piemēram, uzlabojot preču sortimentu, kurā varam ieviest dažādas kategorijas, veikt apmaksāšanu caur dažādiem veidiem un, lai klients būtu spējīgs precīzi izsūtīt caur citiem piedāvātiem piegādātājiem, piemēram, Omniva, DPD u.t.t.

Preču mazumtirdzniecība ir guvusi krietnu popularitāti kopš 2020. gada pandēmijas, jo tas ierosināja vairākus uzņēmumus digitalizēt savu preču mazumtirdzniecību tiešsaistē. Tai skaitā, Latvijā preču mazumtirdzniecība konkurence neatpaliek, jo šeit tā ir pietiekami plaša, piemēram, salīdzinot ar citiem, līdzīgiem e-veikaliem, kā Europick.lv, netprice.lv un rietumprecas.lv. Visiem šiem trīs konkurentiem kaut kas trūkst, tas var būt lapas dizains, SEO u.tml. Mans uzdevums ir pārvērst šos trūkumus par priekšrocībām un ieviest tos savā, pašu veidotā e-veikalā.

2.2. Sistēmas funkcionālās prasības

2.2.1. Klienta sadaļas lapa

P.1. Kontaktu veidlapas aizpildīšana, nosūtot to uz e-pastu, mājaslapas ietvaros

Mērķis:

Sniegt lietotājam iespēju sazināties ar e-veikala administratoriem, aizpildot sadaļā “Kontakti” norādīto veidlapu un nosūtīt to uz attiecināmo e-pastu

Ievaddati:

Lietotājam jāapmeklē sadaļa “Kontakti” un jāaizpilda visi ievades lauki un jānospiež attiecināmā poga

Apstrāde:

Funkcija pārbauda vai lietotājs ir aizpildījis visus nepieciešamos ievades laukus un nospiedis sūtīšanas pogu. Pieņemtie dati tiek nosūtīti uz attiecināmo e-pastu: support@piffdeals.lv

Izvaddati:

E-veikala klienta lietotājam tiek izvadīts sekmīgs paziņojums

P.2. Preču kategoriju ielāde sākuļlapā

Mērķis:

Preču kategoriju ielādi sākuļlapā, galvenes sadaļā

Ievaddati:

E-veikala lietotājam ir jāapmeklē sākuļlapa un administratoram ir jāizveido preču kategorija un jāsavieno tā ar aktīvu grupu

Apstrāde:

Sākuļlapas ielādēšanas procesā, skripts ielādē jaunākos datus par kategorijām. Pārbauda vai kategorijas statuss ir "Aktīvs" un vai vismaz viena grupa ir savienota ar preču kategoriju.

Izvaddati:

Pēc katras lapas pārlādes, skripts atjauninās galvenes sadaļu ar jaunām kategorijām

P.3. Preču grupu ielāde sākuļlapā

Mērķis:

Uzrādīt lapas klientam kategoriju grupas

Ievaddati:

E-veikala lietotājam ir jāapmeklē sākuļlapa un administratoram ir jāizveido vismaz viena preču kategorija, kura ir savienota ar vismaz ar vienu aktīvu grupu

E-veikala lietotājam ar peles kursoru, jānovieto uz vienu no preču kategorijām, lai uzrādītu preču grupas

Apstrāde:

Funkcija, sākuļlapā, ielādē visas aktīvās preču kategorijas, kurām ir viena vai vairākas preču grupas un ielādē izvēlnes sarakstu pēc pieprasījuma

Izvaddati:

Funkcija ielādē preču grupas, kura ir pieejama savā preču kategorijā

P.4. Preču atribūtu ielāde sākuļlapā

Mērķis:

Uzrādīt lapas klientam kategoriju grupas, kurām ir atribūti

Ievaddati:

E-veikala lietotājam ir jāapmeklē sākumlapa un administratoram ir jāizveido vismaz viena preču kategorija, kura ir savienota ar vismaz ar vienu aktīvu grupu, un grupai jāpastāv viens vai vairāki preču atribūti

E-veikala lietotājam ar peles kursoru, jānovieto uz vienu no preču kategorijām, lai uzrādītu preču grupas un to atribūtus

Apstrāde:

Funkcija, sākumlapā, ielādē visas aktīvās preču kategorijas, kurām ir viena vai vairākas preču grupas, kurā zem grupas ir attiecīšie atribūti, un ielādē izvēlnes sarakstu pēc pieprasījuma

Izvaddati:

Funkcija ielādē preču atribūtus, kura ir pieejama savā preču kategorijā, zem savas preču grupas

P.5. Preču skaita uzskaites meklēšanas joslā

Mērķis:

Zinot e-veikala klienta par aktīvu preču skaitu aizvietotāja (*placeholder*) vietā

Ievaddati:

E-veikala klientam ir jāapmeklē mājaslapa. Galvenes josla ir pieejama jebkurā lapā

Apstrāde:

Funkcija atlasa visas preces datubāzē, kam statuss ir "Aktīvs"

Izvaddati:

Lapu pārlādēšanas gadījumā, galvenes meklēšanas joslā tiek ielādēts aktuālā skaita preces

P.6. Preču meklēšana caur meklēšanas joslu

Mērķis:

Nodrošināt e-veikala klientam iespēju meklēt preces caur tiem atslēgasvārdiem pa meklēšanas joslu

Ievaddati:

E-veikala klientam ir jāievada preču atslēgasvārdi, kas atvērtu meklēšanas rezultātu logu

Apstrāde:

Funkcija, pēc atslēgasvārdu ievadišanas, atrod vai neatrod preci un izvada to meklēšanas logā. Funkcija atradīs preci, ja tai statuss ir “Aktīvs”

Izvaddati:

E-veikala klientam tiek atvērta meklēšanas rezultātu logs, kurā tiek uzrādīta meklētā prece (ja tāda pastāv)

P.7. Preču atlase pēc to grupas

Mērķis:

Nodrošināt iespēju filtrēt preces, izvēloties grupu galvenes sadaļā

Ievaddati:

Lietotājam ir ar peles kursoru, jānovieto uz preču kategorijas, kas ieslēgtu izvēlnes sarakstu. Ar peles kreiso klikšķi, jānospiež uz preču grupas

Apstrāde:

Funkcija ielādē jaunu lapu, kurā tiek ir atlasītas preces, kuras pieder pie attiecināmās preču grupas un, kam statuss ir “Aktīvs”

Izvaddati:

Tiek ielādēta jauna lapa ar filtrētiem precēm

P.8. Preču atlase pēc to atribūta

Mērķis:

Nodrošināt iespēju filtrēt pieejamās preces, izvēloties preču atribūtu galvenes sadaļā

Ievaddati:

Lietotājam ir ar peles kursoru, jānovieto uz preču kategorijas, kas ieslēgtu izvēlnes sarakstu. Ar peles kreiso klikšķi, jāuzspiež uz preču atribūta, kas atrodas zem preču grupas

Apstrāde:

Funkcija ielādē jaunu lapu, kurā tiek ir atlasītas preces, kuras pieder pie attiecināmās preču grupas un to grupu atribūta, ja preču statuss ir “Aktīvs”

Izvaddati:

Tiek ielādēta jauna lapa ar filtrētiem precēm

P.9. Izvēlnes saraksts par profilu galvenes joslā

Mērķis:

Piešķirt autorizētam lietotājam piekļuvi izvēlnes sarakstam, uzspiežot uz profila pogas. Izvēlnes saraksta saturs ir autorizētā lietotāja e-pasts un vārds, uzvārds, kā arī citas pogas, piemēram, “ATSLĒGTIES”, “PROFILS”, “GROZS” u.tml.

Ievaddati:

Klientam ir jāreģistrējas sistēmā

Apstrāde:

Veiksmīgas sesijas uzsākšanai, autorizētam lietotājam ir piekļuves izvēlnes sarakstam

Izvaddati:

Autorizēts lietotājs ir tiesīgs atvērt izvēlnes sarakstu, kurā var uzzināt savu e-pastu un vārdu, uzvārdu

P.10. Izvēlnes saraksta saturs lapas administratoram

Mērķis:

Sniegt iespēju lapas administratoram piekļūt administrācijas panelim caur izvēlnes sarakstu galvenes joslā

Ievaddati:

1. Administratoram jābūt autorizētam sistēmā
2. Ar peles kreiso klikšķi, jāatver izvēlnes saraksts
3. Jāuzspiež uz pogas "ADMIN"

Apstrāde:

Tiek ielādēta administrācijas paneļa lapa. Pēc noklusējuma, tā ir pasūtījumu lapa

Izvaddati:

Administrators ir piekļuvis administrācijas panelim

P.11. Profila sadaļas piekļuve

Mērķis:

Sniegt lietotājam iespēju piekļūt sava profila sadaļai.

Ievaddati:

Lietotājam ir jābūt reģistrētam un autorizētam sistēmā. Lietotājam jāuzspiež uz pogas "Profils" izvēlnes sarakstā.

Apstrāde:

Funkcija pārbauda, vai lietotājs ir autorizēts. Ja lietotājs ir autorizēts, tiek ielādēta profila sadaļa, kurā lietotājs var apskatīt un rediģēt savu personīgo informāciju, pasūtījumu vēsturi un citas ar profilu saistītās detaļas.

Izvaddati:

Lietotājs tiek novirzīts uz sava profila sadaļu, kurā var apskatīt un rediģēt savu informāciju.

P.12. Administrācijas paneļa pogas ielāde

Mērķis:

Sniegt administratoram iespēju piekļūt administrācijas panelim caur pogu, kas ielādējas tikai tad, ja lietotājam ir administratora loma.

Ievaddati:

Lietotājam ir jābūt autorizētam sistēmā un jābūt administratora lomai (role_id = 2).

Apstrāde:

Funkcija pārbauda, vai lietotājam ir administratora loma. Ja lietotājam ir administratora loma, administrācijas paneļa poga tiek ielādēta gan galvenes izvēlnē (navbar), gan kājenē (footer). Poga būs redzama tikai tiem lietotājiem, kuriem ir atbilstošās tiesības.

Izvaddati:

Administrācijas paneļa poga parādās galvenē un kājenē. Kad administrators uzspiež uz pogas, tiek ielādēts administrācijas panelis, kurā var pārvaldīt vietnes funkcionalitāti un saturu.

2.2.2. Preču pasūtīšana un preces

P.13. Preču atlase pēc to atribūta

Mērķis:

Lietotājs ir tiesīgs piekļūt lapai, kurā tiek atlasītas preces pēc to atribūta

Ievaddati:

Administratoram ir jāizveido preces, kuras ir pieejamas un sakārtotas, klientam jāuzspiež uz preču atribūta pogas, kas novirza uz preču atlases lapu

Apstrāde:

Nospiežot uz preču atribūta pogas, funkcija atlasa pieejamās preces, kuras ir zem atribūta

Izvaddati:

Lietotājs tiek novirzīts uz preču atribūta atlases lapas

P.14. Preču atlase pēc to grupas

Mērķis:

Lietotājs ir tiesīgs piekļūt lapai, kurā tiek atlasītas preces pēc to grupas

Ievaddati:

Administratoram ir jāizveido preces, kuras ir pieejamas un sakārtotas, klientam jāuzspiež uz preču grupas pogas, kas novirza uz preču atlases lapu

Apstrāde:

Nospiežot uz preču grupas pogas, funkcija atlasa pieejamās preces, kuras ir zem tās grupas

Izvaddati:

Lietotājs tiek novirzīts uz preču grupas atlases lapas

P.15. Preču lapas apskatīšana

Mērķis:

Nodrošināt iespēju lietotājam apskatīt preci no attiecīgajiem logiem

Ievaddati:

Administratoram ir jāizveido preces, kurām statuss ir “Aktīvs” un lietotājam ir jāapmekle lapa, kurā preces ir atlasītas, piemēram, sākulapa, grupu vai atribūtu lapa vai jālieto meklēšanas josla galvenē

P.16. Pieejamā prece tiek ievietota lietotāja grozā

Mērķis:

Lietotājs var sekmīgi ievietot izvēlēto preci savā grozā

Ievaddati:

Administratoram ir jāizveido preces, kuras ir pieejamas un lietotājam ir jāapmeklē konkrētās preces lapa, un jābūt autorizētam

Apstrāde:

Nospiežot pogu “Ievietot grozā”, lietotāja izvēlētā prece tiek ievietota grozā un gan neautorizēts, gan autorizēts lietotājs to var apskatīt.

Izvaddati:

Tiek izvadīts sekmīgs paziņojums un lietotājs var izvēlēties turpināt iepirkšanos vai uzreiz apskatīt grozu

P.17. Preču lapas apskatīšana

Mērķis:

Nodrošināt iespēju lietotājam apskatīt preci no attiecīgajiem logiem

Ievaddati:

Administratoram ir jāizveido preces, kurām statuss ir “Aktīvs” un lietotājam ir jāapmekle lapa, kurā preces ir atlasītas, piemēram, sākulapa, grupu vai atribūtu lapa vai jālieto meklēšanas josla galvenē

Apstrāde:

Uzspiežot uz preces kartiņas, lietotājs tiek novirzīts uz preču lapas ar URL /good{id}

Izvaddati:

E-veikala klients ir novirzīts uz preču apskates mājaslapu, kurā tiek informēts par preču saturu un ir tiesīgs pievienot preci grozam

P.18. Preču skaita modificēšana grozā

Mērķis:

Nodrošināt iespēju lietotājam mainīt pasūtījamo preču skaitu grozā

Ievaddati:

Lietotājam jāievieto preces grozā un jāapmeklē grozas lapa

Apstrāde:

Uzspiežot uz preču skaita ievades lauka, lietotājs var mainīt preču skaitu

Izvaddati:

Uzspiežot uz pogu “PASŪTĪT”, datubāze atjaunina preču skaita datus un novirza lietotāja uz pasūtījuma nomaksas lapu

P.19. Pasūtījuma e-pasta vēstule

Mērķis:

Nodrošināt iespēju lietotājam saņemt rēķinu lietotāja konta e-pasta adresē

Ievaddati:

Lietotājam ir jāpasūta preces un jāaizpilda kontaktinformācija

Apstrāde:

Uzspiežot uz pogas “PASŪTĪT”, tiek izveidoti jauni dati tabulā “Adresses” un “Orders”

Izvaddati:

Lietotājs saņem pasūtījuma čeku e-pasta adresē

P.20. Jaunu preču izveide

Mērķis:

Nodrošināt iespēju administratoram izveidot preces

Ievaddati:

Administratoram jāapmeklē */admin-goods* lapa un jāuzspiež uz pogas “Pievienot preci” un jāaizpilda forma

Apstrāde:

Uzspiežot uz pogas “Pievienot preci”, funkcija izveido datubāzē preču šablonu

Izvaddati:

Pēc preču šablonu izveidošanas brīdī, administrators tiek novirzīts uz preču rediģēšanas lapu

P.21. Eksistējošu preču rediģēšanu

Mērķis:

Nodrošināt iespēju administratoram rediģēt eksistējošas preces

Ievaddati:

Administratoram jāapmeklē */admin-goods* lapa un jāuzspiež uz pogas “Pievienot preci” un jāaizpilda forma, izveidošanas gadījumā administrators tiek novirzīts uz preču rediģēšanas lapu

Apstrāde:

Funkcija pārbauda formas datus un atjaunina preces datus datubāzē

Izvaddati:

Izvada paziņojumu par preču rediģēšanu un novirza lietotāju atpakaļ */admin-goods* lapā

2.2.3. Lietotāja kots

P.22. Konta pieslēgšanās

Mērķis:

Veikt sekmīgu konta pieslēgšanos e-veikala ietvaros vietās, kurās tas tiek pieprasīts

Ievaddati:

Lietotājam ir jāreģistrē sev kots. Lietotājam ir jāapmeklē sadaļa “Profils” vai jāpanāk solis, kad pieprasa pieslēgšanos e-veikalā, piemēram, pievienojot preci grozam u.tml.

Apstrāde:

Funkcija pārbauda visus ievadītos datus no lietotāja un secina, lai uzsāktu sesiju un pieslēgtu lietotāju e-veikalā

Izvaddati:

Lietotājs tiek paziņots par sekmīgu pieslēgšanos un tiek novirzīts uz e-veikala sākumlapu

P.23. Konta reģistrācija

Mērķis:

Veikt sekmīgu konta reģistrāciju e-veikala ietvaros

Ievaddati:

Lietotājam ir jāapmeklē sadaļa “Profils” vai jāpanāk solis, kad pieprasa pieslēgšanos e-veikalā, pēc tam nospiežot pogu “Reģistrēties”

Apstrāde:

Funkcija pārbauda vai visi ievadītie lauki ir aizpildīti un reģistrē kontu datubāzē

Izvaddati:

Lietotājs tiek paziņots par sekmīgu reģistrāciju un tiek novirzīts uz e-veikala sākumlapu

P.24. Paroles šifrēšana, reģistrējot kontu

Mērķis:

Veikt sekmīgu paroles šifrēšanu drošības nolūkiem, veicot konta reģistrāciju procesā

Ievaddati:

Lietotājam ir jāapmeklē sadaļa “Profils” vai jāpanāk solis, kad pieprasa pieslēgšanos e-veikalā, pēc tam nospiežot pogu “Reģistrēties”, un sekmīgi jāizveido konts aizpildot visus nepieciešamos laukus

Apstrāde:

Funkcija pārbauda vai visi ievadītie lauki ir aizpildīti un reģistrē kontu datubāzē. Pirms konta datu ievietošanas datubāzē, tiek lietots *Laravel Bcrypt*, kas šifrē paroli 60 simbolu garumā

Izvaddati:

Lietotāja konta parole tiek sekmīgi šifrēta un ievietota datu bāzē

P.25. Paroles atjaunošanas vēstule

Mērķis:

Nodrošināt neautorizētam lietotājam iespēju pieprasīt paroles atjaunošanas e-pastu pieslēgšanās sadaļā.

Ievaddati:

Neautorizētam lietotājam ir jāapmeklē pieslēgšanās sadaļa un jāuzspiež uz saites "Aizmirsi paroli?" un jāievada derīgs e-pasts.

Apstrāde:

Funkcija pārbauda, vai ievadītais e-pasts pastāv datubāzē. Ja e-pasts pastāv, tiek nosūtīts paroles atjaunošanas e-pasts.

Izvaddati:

Lietotājs saņem paroles atjaunošanas e-pastu ar saiti, lai atjaunotu paroli.

P.26. Paroles atjaunošana saite

Mērķis:

Nodrošināt lietotājam iespēju atjaunot paroli pēc paroles atjaunošanas e-pasta saņemšanas.

Ievaddati:

Lietotājam ir jāatver paroles atjaunošanas e-pasts un jāuzspiež uz saites "Atjaunot paroli", pēc tam jāaizpilda forma, kurā ievades lauks automātiski tiek aizpildīts ar tā e-pastu (e-pasta lauks ir deaktivizēts), un divi paroles ievades lauki "ievadi paroli" un "ievadi paroli atkārtoti".

Apstrāde:

Funkcija pārbauda, vai abas ievadītās paroles ir vienādas. Ja paroles sakrīt, lietotāja konta parole tiek nomainīta un e-pasta ievades lauks nav kompromizēts

Izvaddati:

Lietotājs tiek informēts par veiksmīgu paroles atjaunošanu un tiek novirzīts uz pieslēgšanās sadaļu.

P.27. Konta reģistrācija, lietojot Google OAuth API

Mērķis:

Nodrošināt lietotājiem iespēju reģistrēt savu kontu, izmantojot Google API.

Ievaddati:

Lietotājam jāapmeklē reģistrācijas lapa un jāuzspiež uz pogas "Reģistrēties ar Google".

Apstrāde:

Funkcija nosūta pieprasījumu uz Google API, lai iegūtu nepieciešamos lietotāja datus. Pēc veiksmīgas autentifikācijas, funkcija saņem lietotāja datus no Google API un piešķir sesijas kodu.

Izvaddati:

Lietotājs tiek sekmīgi reģistrēts un lietotājs tiek novirzīts uz e-veikala sāukmlapu ar piešķirtu sesijas kodu.

P.28. E-pasta verifikācija

Mērķis:

Nodrošināt lietotājam tiesības apstiprināt savu e-pasta adresi

Ievaddati:

Lietotājiem jāapmeklē konta reģistrācijas lapa un jāreģistrē konts

Apstrāde:

Funkcija, pēc noklusējuma, aizsūta lietotājam vēstuli uz e-pasta adresi pēc reģistrācijas vai lietotājam jāapmeklē */verify-email* lapa, un jāpieprasa jauna vēstule e-pasta adrese verificēšanai.

Funkcija pārbauda vai e-pasta ir verificēta, ja nav, tad sāukmlapā izvada atgādinājumu par e-pasta verificēšanu

Izvaddati:

Lietotājs ir verificējis e-pastu un tagad ir tiesīgs pasūtīt preces

P.29. Atcerēties mani funkcionalitāte

Mērķis:

Nodrošināt lietotājam iespēju izvēlēties “Atcerēties mani” funkcionalitāti, kas saglabā pieslēgšanās stāvokli pēc mājaslapas aizvēršanas.

Ievaddati:

Lietotājam jāatķeksē izvēles rūtiņa “Atcerēties mani” pieslēgšanās lapā un jāveic pieslēgšanās.

Apstrāde:

Funkcija izveido “remember me” token tipa atslēgu un saglabā to lietotāja tabulā datubāzē. Token atslēga tiek pievienots lietotāja sesijai, ļaujot pieslēgšanās stāvokli saglabāt pēc mājaslapas aizvēršanas.

Izvaddati:

Lietotājam tiek sekmīgi piešķirta “Atcerēties mani” funkcionalitāte, un lietotājs vairs netiek pieprasīts autentifikācijas, atkārtoti apmeklējot mājaslapu.

P.30. Konta informācijas atjaunināšana

Mērķis:

Sniegt lietotājam iespēju atjaunināt savu profila vārdu, uzvārdu un e-pastu

Ievaddati:

Lietotājam jābūt autorizētam un jāapmeklē sadaļa “Profils”. Jāaizpilda atjaunināšanas forma ar jauniem datiem un jāapstiprina izmaiņas.

Apstrāde:

Funkcija pārbauda, vai visi ievadītie lauki ir aizpildīti un derīgi. Ja dati ir derīgi, funkcija atjaunina lietotāja informāciju datubāzē.

Izvaddati:

Lietotājs saņem paziņojumu par veiksmīgu konta informācijas atjaunināšanu un redz jaunus datus savā profilā.

P.31. Lietotāja konta dzēšana

Mērķis:

Nodrošināt lietotājam iespēju dzēst savu kontu.

Ievaddati:

Lietotājam jābūt autorizētam un jāapmeklē sadaļa “Profils”. Jāuzspiež uz pogas “Dzēst kontu” un jāapstiprina darbība.

Apstrāde:

Funkcija pārbauda, vai lietotājs ir autorizēts un apstiprinājis konta dzēšanu. Ja apstiprināts, funkcija pārtrauc pašreizējo sesiju un anonimizē lietotāja konta datus datubāzē

Izvaddati:

Lietotājs saņem paziņojumu par veiksmīgu konta dzēšanu un tiek novirzīts uz mājaslapas sākumlapu kā viesis.

P.32. Lietotāja pasūtījumu vēstures apskate

Mērķis:

Nodrošināt lietotājam iespēju apskatīt savu pasūtījumu vēsturi.

Ievaddati:

Lietotājam jābūt autorizētam un jāapmeklē sadaļa “Pasūtījumu vēsture”, kura atrodas sadaļā “Profils”.

Apstrāde:

Funkcija atlasa un ielādē visus lietotāja pasūtījumus no datubāzes, kuri saistīti ar lietotāja kontu.

Izvaddati:

Lietotājs redz savu pasūtījumu vēsturi ar informāciju par katru pasūtījumu, ieskaitot datumu, preces un statusu.

P.33. Lietotāja konta paroles maiņa

Mērķis:

Nodrošināt lietotājam iespēju mainīt savu konta paroli.

Ievaddati:

Lietotājam jābūt autorizētam un jāapmeklē sadaļa “Profils” un jāuzspiež uz paroles maiņas pogas, un jāaizpilda forma

Apstrāde:

Funkcija pārbauda vai pašreizējā parole sakrīt ar datubāzes ierakstu, funkcija pārbauda vai jaunās paroles ievades lauki sakrīt un tikai pēc pogas nospiešanas, atjaunina datus datubāzē

Izvaddati:

Tiek izvadīts paziņojums par paroles maiņu

2.2.4. Administrācijas panelis

P.34. Administratora konta piekļuve lapām

Mērķis:

Nodrošināt, ka tikai administratora konti var piekļūt administrācijas lapām, lai uzlabotu sistēmas drošību.

Ievaddati:

1. Lietotājam ir jābūt autorizētam sistēmā
2. Lietotājam jābūt administratora lomai

Apstrāde:

Sistēma pārbauda lietotāja lomu pēc sesijas sākuma.

Ja lietotājam nav administratora lomas, tiek parādīta kļūdas lapa (Error 404) anonimitātes nodrošināšanai, nevis parasta kļūda (Error 403).

Izvaddati:

Administratora konti var piekļūt administrācijas lapām.

Lietotāji bez administratora lomas redz kļūdas lapu (Error 404).

P.35. Administratora mega meklēšanas josla

Mērķis:

Nodrošināt administratoriem iespēju izmantot mega meklēšanas joslu, lai ātri atrastu dažādus sistēmas elementus, piemēram, pasūtījumu, lietotāja kontu, preci

Ievaddati:

1. Administratoram jābūt autorizētam un jāpiekļūst administrācijas panelim.
2. Jāievada meklēšanas atslēgas vārdi meklēšanas joslā.

Apstrāde:

Meklēšanas josla apstrādā ievadītos atslēgas vārdus.

Meklēšanas algoritms meklē atbilstības dažādās datu bāzes tabulās, piemēram, pasūtījumu numuri, preču nosaukumi, lietotāju e-pasti, lietotāju tālrunu numuri

Izvaddati:

Tiek parādīti meklēšanas rezultāti, kas atbilst ievadītajiem atslēgas vārdiem.

P.36. Administratora paziņojumi

Mērķis:

Nodrošināt paziņojumu sistēmu, kas informē administratorus par pasūtījumu statusiem.

Ievaddati:

Pasūtījumu statusi: "procesā", "atcelts", "pabeigts", "gaida".

Administratoram jābūt autorizētam sistēmā.

Apstrāde:

Paziņojumu sistēma ģenerē paziņojumus, pamatojoties uz pasūtījumu statusiem.

Paziņojumu logs rāda nesenos paziņojumus.

Divas pogas paziņojumu logā: "skatīt visus" un "atzīmēt kā izlasītus".

Poga "skatīt visus" ielādē atsevišķu lapu, kurā parādīti visi paziņojumi.

Poga "atzīmēt kā izlasītus" iztīra paziņojumu logu un atzīmē paziņojumus kā izlasītus.

Izvaddati:

Administratoram tiek parādīti paziņojumi par pasūtījumu statusiem.

Paziņojumu logs tiek iztīrīts pēc pogas "atzīmēt kā izlasītus" nospiešanas.

P.37. Administratora konta vārds, uzvārds

Mērķis:

Parādīt administratora konta vārdu un uzvārdu galvenē (navbar), lai uzlabotu lietotāja pieredzi un nodrošinātu identifikāciju.

Ievaddati:

Administratoram jābūt autorizētam sistēmā.

Apstrāde:

Sistēma no datu bāzes ielādē administratora vārdu un uzvārdu.

Administratora vārds un uzvārds tiek parādīts galvenē (navbar).

Izvaddati:

Galvenē tiek parādīts autorizētā administratora vārds un uzvārds.

P.38. Kataloga komponentes

Mērķis:

Nodrošināt administratoriem iespēju pārvaldīt preču katalogu ar datu tabulu (datatable) palīdzību, ļaujot pievienot, dzēst, rediģēt, filtrēt un lietot lappušu numerācijas funkcijas un skatīt tabulas elementus klienta pusē.

Ievaddati:

Administratoram jābūt autorizētam un jāapmeklē attiecīgā sadaļa administrācijas panelī (piemēram, kategorijas, grupas, atribūti un preces).

Apstrāde:

Katras sadaļas lapā tiek ielādēta datu tabula ar pieejamajiem datiem. Administratori var izmantot lappušu numerāciju funkcijas, meklēšanas joslu un filtru funkcijas, lai pārvaldītu un atlasītu datus pēc aktīvām kategorijām un grupām.

Izvaddati:

Administratori var pievienot, dzēst, rediģēt un skatīt tabulas datus, kā arī preces klienta pusē. Datutabulā tiek atspoguļoti aktuālie rezultāti ar lappušu numerāciju funkcijām un filtrēšanas iespējām.

P.39. Klientu konta administrācija

Mērķis:

Nodrošināt administratoriem iespēju pārvaldīt klientu reģistrētā konta datus, sūtīt paroles atjaunošanas e-pastus un rediģēt kontus.

Ievaddati:

Administratoram jābūt autorizētam un jāapmeklē sadaļa “Klienti” administrācijas panelī.

Apstrāde:

Sadaļā tiek ielādēta datu tabula ar reģistrēto klientu datiem. Administratoriem ir iespēja izmantot meklēšanas joslu, lappušu numerāciju funkcijas un rediģēt klientu kontus. Administratori var izveidot jaunus klientu kontus un sūtīt paroles atjaunošanas e-pastus, kā arī dzēst kontus no datubāzes.

Izvaddati:

Administratori var pārvaldīt klientu kontus, rediģēt tos un sūtīt paroles atjaunošanas e-pastus. Datu tabulā tiek atspoguļoti aktuālie klientu dati ar pagināciju un filtrēšanas iespējām.

P.40. Klienta konta dzēšana administrācijas panelī

Mērķis:

Nodrošināt administratoram pieeju dzēst klienta kontus no datubāzes

Ievaddati:

Administratoram jāapmeklē klientu sadaļa un tabulā, jāuzspiež uz apskates pogas, un jāuzspiež uz pogas “Dzēst kontu”

Apstrāde:

Administrators veic apstiprinājumu konta dzēšanai. Funkcija pārbauda pieprasījumu un dzēš kontu no datubāzes

Izvaddati:

Tiek izvadīts paziņojums par veiksmīgu konta dzēšanu datubāzē

P.41. Klienta konta paroles atiestatīšanas funkcionalitāte

Mērķis:

Nodrošināt administratoram pieeju izsūtīt paroles atiestatīšanas vēstuli klienta kontam

Ievaddati:

Administratoram jāapmeklē */admin-users* lapa un uz attiecīgā klienta tabulā, jāuzspiež uz paroles atiestatīšanas pogas

Apstrāde:

Funkcija sagaida administratora apstiprinājumu un uz klienta konta e-pastu nosūta paroles atiestatīšanas vēstuli

Izvaddati:

Tiek izvadīts paziņojums par veiksmīgu paroles atiestatīšanas vēstuli nosūtīšanu

2.2.5. Citi

P.42. Augšup ritināšanas poga

Mērķis:

Sniegt lietotājam iespēju pēc pogas klikšķa, automātiski ritināt līdz pašai augšai lapas ietvaros

Ievaddati:

Lietotājam ir ar peles ritināšanas rullīti, jānotin lapa uz leju vismaz 25% no visa augstuma, lai ieslēgtos ritināšanas poga un ar peles kreiso klikšķi uz tā janospiež, lai automātiski uzritinātu līdz pašai augšai

Apstrāde:

Funkcija pārbauda vai lapa ir noritināta uz leju vismaz 25% no visa lapas augstuma, lai būtu spējīgs ieslēgt ritināšanas pogu, kura atrodas labā, apakšējā stūrī ar *CSS absolute* pozīcijas variantu un ar peles kreiso klikšķi nospiešanas gadījumā, tā uzritinātu lietotāju līdz lapas pašai augšai

Izvaddati:

Lietotājs atrodas lapas augšā

P.43. Datu šifrēšana, lietojot OpenSSL algoritmu

Mērķis:

Uzlabot datubāzes drošības stāvokli, šifrējot visus ienākošos un izejošos datus

Ievaddati:

Lapas lietotājiem ir jāveic preču iegāde, jaunu preču izveidošana, konta reģistrācija

Apstrāde:

Funkcija šifrē visus datus pirms sūtīšanas un atšifrē visus ienākošos datus no datubāzes

Izvaddati:

Dati tiek šifrēt sūtīt un atlasot no datubāzes

2.3. Sistēmas nefunkcionālās prasības

1. Lietotāju un vietnes saskarne pēc noklusējuma ir latviešu valodā;
2. E-veikala tīmekļa vietnes ir paredzēta strādāt uz mūsdienīgu interneta pārlūkprogrammām;
3. E-veikala dominējošās krāsas ir spilgti, elektro zila, sekundārā krāsa ir ar gaišāku toni, elektro zila un akcenta krāsa ir vēl lielāka, gaišāku toni. Krāsu *HEX* kodus var uzzināt tikai pēc pieprasījuma;
4. Domēns priekš e-veikala ir <https://piffdeals.lv>, tam piekļūt var caur pārlūkprogrammas adreses joslas logu;
5. Tīmekļa vietnes mitināšanas pakalpojuma (*web host*) sniedzējs nes atbildību par mājaslapas uzturēšanu uz servera, tas nozīmē, ka vietnes uzturētājam ir jāsaņemas ar pakalpojuma sniedzēju gadījumā, ja vietnē ir servera problēmas;
6. Tīmekļa vietnes izstrādātāji un uzturētāji nav spējīgi prognozēt servera kļūdas, kurus izveido vietnes mitināšanu (*web hosting platform*) platforma sniedzēji.

2.4. Gala lietotāja raksturiezīmes

“Piffdeals.lv” e-veikals kalpo kā mugurkauls uzņēmuma preču mazumtirdzniecības biznesā. Tas radikāli atvieglo mana klienta dzīvi, jo pašlaik klients savas preces reklamē caur citu sociālu tīklu platformām, piemēram, Telegram un Whatsapp u.t.t.

Tīmekļa vietnes mērķauditorija ir ikdienas iedzīvotāji, kuri vēlas veikt iepirkšanos tiešsaistē – pasūtot dažāda veida preces, kas virzās no sadzīves ķīmijas līdz pat ēdienu saldumiem.

Vietnes klienti ir spējīgi veikt preču pasūti, veicot pirms tam konta reģistrāciju un verificējot e-pasta adresi, šī funkcionalitāte ir nepieciešama, lai novērstu ļaunprātīgu pasūtījumu izveidi.

Pirmās mājaslapas versijas ietver sniegt iespēju apskatīt preces un veidot pasūtījumus, neveicot apmaksu. Vietnē tiek lietots primitīvs dizains, kurš nav mūsdienisks, tāpēc tas tiks mainīts. Izstrādātājiem jāparūpējās par klienta jeb lietotāja pieredzi, mazinot peles klikšķus, tas nozīmē, ka nav jāveido lieki klikšķi priekš vienas darbības. Nākotnes plāni iekļauj mainīt e-veikala infrastruktūru preču pasūtīšanai, jo tiks ieviesti dažādi apmaksas veidi un klienti varēs pasūtīt preces pakomātos.

3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

Izstrādes līdzekļu un rīku sadaļa tiek sastādīta, lai informētu lasītāju ar kādiem līdzekļiem un pielietotiem rīkiem tiek veidots projekts. Tiek analizēti izstrādātāju pielietotie līdzekļi un sniegts atbilstošs pamatojums. Izstrādātājs izpētījis jeb analizējis citu alternatīvu iespējamus variantus kā vēl izstrādāt e-veikalu. Par katru līdzekli un rīku tiek sniegts apraksts un beigās kopsavilkuma pamatojums. Sadaļā var tikt lietoti sveši termini, kuri būs sīkāk aprakstīti sadaļā “Lietoto saīsinājumu un terminu skaidrojums”. Visiem lietotiem līdzekļiem un rīkiem pastāv sava mājaslapa, kura būs norādīta aprakstā.

3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

Projekts tiek izstrādāts, pielietojot VILT steku. VILT steks sastāv no *Vue*, *Inertia*, *Laravel*, *Tailwind*. Pastāv vairāki veidi kā izstrādāt savu e-veikala projektu. Izstrādātājs ir izvēlējis šādus līdzekļus un rīkus, un to iemesli ir argumentēti:

3.1.1. Back-end puses programmēšanas valodas līdzekļi

- **Laravel**

Populārs PHP programmēšanas valodas bāzēts *framework*, kas sniedz izstrādātājam plašu klāstu caur to MVC arhitektūru, drošības pasākumiem un kopienas atbalstu. Laravel lieto *Eloquent ORM*, kas atvieglo darbības saistībā ar datu bāzēm, piemēram, veidot drošus SQL vaicājumus u.tml. Iekš Laravel tiek veikti bieži atjauninājumi saistībā ar drošību, piemēram, drošība sakarā ar CSRF, SQL injekcijām un XSS tipa uzbrukumiem vienmēr būs labāk uzlabota nekā parastā PHP programmēšanas valodā. Sīkāka informācija: laravel.com

- **MySQL**

Datu bāzes relācijas pārvaldības sistēma. Izcils variants maziem un lieliem projektiem, kurā vēl sniedz vienkāršu un elastīgu sintaksi. Ļoti atbalstīts no mitināšanas pakalpojuma sniedzējiem (*web hosting platforms*). MySQL datu bāzes kodols ir tās tabulas, kura sastāv no rindām un kolonnām. Sīkaka informācija: mysql.com

- **XAMPP**

XAMPP ir brīva, atvērta koda (*open-source*) tīmekļa vietņu serveris. XAMPP ir ļoti populārs starp izstrādātājiem, it īpaši tiem, kuri ikdienā strādā ar PHP programmēšanas valodu. E-veikala projektā tas tiek izmantots, lai veiktu darbības datu bāzē un nepieciešamos testus lokāli caur to piedāvāto rīku phpMyAdmin pirms izvietojšanas produkcijā uz oficiālu serveri. XAMPP

piedāvā veikt darbības ne tikai ar MySQL, bet arī ar MariaDB. Sīkaka informācija: apachefriends.org

- **Ziggy**

Ziggy ir GitHub repozitorijas pakotne, kura pieejama tikai Laravel. Tā sniedz iespēju lietotājiem lietot Laravel veidotās saites jeb maršrutēšanas pasākumus (*routes*) tieši JavaScript kodā, šai gadījumā tas būs iekš VueJS. Ziggy skaitās back-end pakotne, jo tā uzlabo saikni priekš InertiaJS starp back-end un front-end pusēm. Ziggy ir nesen ieguvis lielu popularitāti starp Laravel izstrādātājiem, jo tas atvieglo saites pārvaldību starp Laravel un VueJS, padarot dzīvi izstrādātājiem vieglāku, īpaši izstrādājot ar SPA un dinamisku tīmekļa aplikāciju izstrādē. Ziggy ir adaptīvs arī citās JavaScript *framework* valodās. Sīkaka informācija: github.com/tighten/ziggy

- **NodeJS jeb/un npm**

Node.js var lietot gan back-end, gan front-end. Projektā tas tiek lietots back-end pusē caur npm (*node package manager*). Tas ir Node.js pakotņu pārvaldnieks, kas atvieglo pakotņu koplietošanu un atkārtotu izmantošanu starp JavaScript projektiem. Laravel projektu nav iespējams palaist bez Node.js, toties var izlaist Vite. Izstrādātāji lieto npm, lai veiktu citu bibliotēku, rīku un API lejupielādi, atjaunināšanu un pārvaldīšanu atkarībā, kas nepieciešams projektā. Node.js spēlē būtisku lomu, lietojot VILT programmēšanas steku, jo tas nodrošina izstrādes vidi priekš VueJS un Tailwind CSS, ļaujot tam darbināt to rīkus un kompilatorus. Node.js nodrošina back-end infrastruktūru un npm ir, kas ļauj pārvaldīt caur šīm pakotnēm. Sīkaka informācija: nodejs.org

3.1.2. Front-end puses programmēšanas valodas līdzekļi

- **VueJS**

VueJS jeb Vue.js vai Vue ir atvērta koda JavaScript *framework*, ko parasti lieto, veidojot lietotāja saskarnes un SPA izstrādi. Vue.js spēlē nozīmīgu ietekmi uz front-end izstrādi, piemēram, caur izstrādātāju pieredzi, jo tam piemīt komponentu bāzēta arhitektūra, līdzīgi kā React, padarot vieglu saskarnes izstrādi un to uzturēšanu, plašs kopienas atbalsts un tās ekosistēma, tas nozīmē, ka kopiena piedāvā plašus spraudņus jeb *plugins*, bibliotēkas un dažādus rīkus. Gūtā informācija no izstrādātāju aptaujām (GitHub, npm, StackOverflow) liecina, ka Vue.js turpina būt viens no populārākajiem JavaScript *framework*, kura cieši konkurē ar React un Angular. Sīkaka informācija: vuejs.org

- **Tailwind CSS**

Tailwind ir CSS *framework*, kuru izmanto pirmo pieeju stila veidošanai, tas ļoti atvieglo izstrādātāju dzīvi, kas ļauj ātri rakstīt CSS kodu, neveidojot atsevišķas stila lapas. Tailwind tika

izveidots, lai primāri paātrinātu front-end darbu, nodrošinot izstrādātājiem lietot stilus iekš HTML definētām klasēm. Tailwind ir kļuvis par vienu no CSS vadošajiem *framework*, pateicoties tā pieejai, kas atšķiras no citiem *frameworks*. Tailwind piedāvā instrumentus jeb klases no CSS puses, kuras jau ir definētas. Viens no Tailwind raksturojošiem elementiem ir tā responsivitāte un adaptīvais dizains priekš mazāk ekrāna izmēra ierīcēm. Tailwind ir brīvs *framework* un to var mierīgi lejupielādēt, lietojot npm. Sīkāka informācija: tailwindcss.com

- **InertiaJS**

Inertia jeb InertiaJS ir JavaScript bibliotēka, kas tēmēta uz SPA izstrādi. Tas ļauj front-end un back-end savienot bez nepieciešamības lietot API kā starpnieku, tas nozīmē, ka var lietot Laravel, Rails vai Django ar VueJS, React vai Svelte un neuztraukties par API lietošanu. Inertia tiek aktīvi izmantots Laravel ietvaros un guvis popularitāti kā alternatīvs, tradicionāls priekš SPA izstrādes metodēm. Inertia ne tiek veido saikni, bet uzlabo front-end veiktspēju vietnes lietotājiem, kas pāatrina lapu ielādi. Sīkāka informācija: inertiajs.com

- **Vite**

Vite ir mūsdienīgs front-end izstrādes rīks, kuru pārsvarā lieto priekš servera konfigurāciju, padarot tās konfigurēšanu elastīgāku, tas nozīmē, ka mājaslapas produkcijas procesā, Vite spēs nodrošināt moduļu aizstāšanu. Evan You, VueJS izstrādātājs, secina, ka galvenā atšķirība no citiem rīkiem kā Webpack, ir tā, ka Vite izmanto ES moduļus tieši pārlūkprogrammas laikā, Vite nodrošina pāatrinātu moduļu ielādi un spēju atjaunināt veiktās izmaiņas. Sīkāka informācija: vitejs.dev

3.1.1. Izmantotie rīki

- **Visual Studio Code**

VS Code ir bezmaksas, atvērtā koda teksta redaktors, kuru izstrādāja Microsoft. Tiek piedāvāta plaša funkcionalitāte, tostarp atbalstu uz dažādām programmēšanas valodām, piemēram, integrētu termināli, versiju kontroles sistēmu Git, lejupielādēt kopienas radītos spraudņus u.c. Daudzpusības dēļ, VS Code ir kļuvis par vienu no populārākajām teksta redaktoriem (*IDE*) programmētāju vidū. Pēc aptauju datiem, VS Code bieži tiek atzīts par vienu no visvairāk mīlētajiem rīkiem pasaulē. VS Code ir universāla izvēle starp dažādu veidu projektu izstrādēm. Tā spēj piedāvāt augstu pielāgojamības un efektivitātes līmeni, kopā ar draudzīgu un elastīgu saskarni. Sīkāka informācija: code.visualstudio.com

- **phpMyAdmin**

Bezmaksas rīks, kas izstrādāts PHP valodā, lai pārvaldītu MySQL un MariaDB datu bāzes caur pārlūkprogrammas saskarni, tas ļauj lietotājiem lietot grafisko lietotāja saskarni, nevis

komandrindas interfeisu caur termināli. phpMyAdmin plaši tiek izmantots vidēja un maza mēroga projektos. Izcils variants jauniem izstrādātājiem, kuri sākuši nodarboties ar datu bāzēm. E-veikala projekta ietvaros, phpMyAdmin tiek pielietots caur XAMPP un arī, lai veiktu datu bāzes darbības caur Laravel *Eloquent ORM*. Rīks ir pieejams uz lielākā daļu mitināšanas pakalpojumiem, padarot to efektīvi sasniedzamu. Phpmyadmin.net

- **Laravel Telescope**

Elegants atklūdošanas (*debugging*) rīks no Laravel izstrādātājiem. Pieeja ir gluži tāda pati kā XAMPP. Laravel Telescope ir pakotne, kuru lejupielādē npm, tās mērķis ir nodrošināt efektīvu uzraudzīšanu un atklūdošanu programmatūras darbības reālā laikā. Tas nozīmē, ka var pētīt informācijas sūtīšanu, piemēram, starp SQL vaicājumiem, citi izņēmumi, e-pasta ziņojumi u.t.t. Laravel Telescope ir iecienīts starp Laravel izstrādātājiem, jo tas ir būtisks rīks, lai identificētu un novērstu problēmas izstrādes un produkcijas laikā. E-veikala ietvaros, izstrādātājs, lieto Laravel Telescope, lai meklētu sistēmas drošības caurumus caur HTTP pieprasījumiem, kā arī veikt sekmīgas SQL vaicājuma izpildes un kontrolēt e-pasta ziņojumus e-veikala klientiem. Sīkāka informācija: laravel.com/docs/10.x/telescope

3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts

- **React**

React ir JavaScript bibliotēka, daudzi mēdz uzskatīt to par ietvaru (*framework*), bet tā nav, salīdzinot ar VueJS, kas ir *framework*. React izstrādāja Facebook, lai veidotu vienlapas programmatūras (*SPA*). React ir domāts liela mēroga tīmekļa aplikācijām, kas var dinamiski atjaunināt un parādīt datus, nepārlādējot lapu. React arī izmanto komponentu bāzētu arhitektūru, kas ļauj atkārtoti izmantot UI komponentus. Sīkāka informācija: react.dev

- **SQLite**

SQLite ir viegla, relācijveida datu bāzes pārvaldības sistēma, kas nesniedz klienta-servera datu bāzes dzinēja funkcionalitāti. Atšķirībā no tradicionālajām datu bāzes sistēmām, SQLite glabā visu datu bāzi kā vienu failu uz diska. Tas ļauj viegli integrēt SQLite jebkurā lietojumprogrammā bez nepieciešamības pēc atsevišķa datu bāzes servera. SQLite ir plaši pazīstama ar savu uzticamību, efektivitāti un neatkarību no platformas. SQLite ietvaros nav nepieciešams uzstādīt datu bāzes serveri un parasti ir domāts mazāk līdz vidēja lieluma programmatūrās. SQLite nav vajadzīgs XAMPP, to var lejupielādēt kā spraudni caur VSCode un tur pat uz vietas palaist. SQLite ir ideāls variants portfolio vai mājaslapām kam ir nepieciešama maza back-end izstrāde. Datu bāzes konfigurācija ir ļoti vienkārša, piemēram, lai savienotu ar

Laravel, tai nav nepieciešams veidot datu bāzes kontu, tai skaitā paroli u.tml., tai vajadzīgs tikai lokāls savienojums. Plašāka informācija šeit: sqlite.org

- **PHP**

PHP ir back-end skriptu, programmēšanas valoda, kas ir domāta tīmekļa vietņu izstrādei. Tā ļauj veidot dinamiskas tīmekļa lapas un veikt savienojumus ar datu bāzēm un citiem API pakalpojumiem, piemēram, Stripe. PHP ir viena no pamata tehnoloģijām tīmekļa izstrādē, un tās ir pamats daudzām tīmekļa lietojumprogrammām un ietvariem, tostarp Laravel. Lai gan Laravel ir uz PHP bāzēts *framework*, tas nevar būt tiešs aizstājējs, drīzāk, Laravel piedāvā kā papildus slāni, piemēram, citus rīkus, kas varētu atvieglot PHP tīra koda rakstīšanu konkrētām vajadzībām. PHP var būt piemērotāks uz vienkāršām tīmekļa lapām vai projektu prototipēšanai, kad izstrādātājam ir nepieciešama pilna kontrole pār savu rakstīto kodu. Sīkāka informācija: php.net

3.3. Līdzekļu un rīku lietojuma pamatojums

Pastāv vairāki varianti kā realizēt e-veikala projektu, tas ir noteikts no paša projekta specifikas un uz labāko praksi jeb tendencēm mūsdienās. Izstrādātājs ir izvēlējies šādu VILT steku, jo tas ir pielietots praksē un no tā ir gūta pieredze. Tomēr, dēļ šāda apjoma projekta līmeņa, ir nācies meklēt alternatīvus variantus, piemēram, plānojumā bija pielietot SQLite, nevis MySQL, jo SQLite ir vienkārša pārvaldības sistēma, kuras interfeiss atrodas pašā VSCode. Tomēr, tika lietots XAMPP un MySQL, jo SQLite ir domāts mazāk apjoma projektiem, piemēram, portfolio tipa darbiem un nav stabils pret e-veikala apjoma tipa mājaslapām.

Tiek lietots Laravel, jo tam ir vienkārša *Eloquent* sintakse saistībā ar datu bāzēm, kaut PHP programmēšanas valodai ir lielāks kopienas atbalsts, tomēr Laravel labi sadarbojas ar VueJS.

Laravel un Vue labāk sadarbojas nekā Laravel un React, piemēram, Laravel piedāvā plašāk sarakstītu dokumentāciju priekš Vue, nevis React. Tika apdomāta stratēģija, kā panākt front-end un back-end komunikāciju, tai vietā nāk InertiaJS, kas palīdz komunicēt starp abām pusēm un padot tālāk precīzu informāciju. Ziggy tika pielietots praksē, kurā abu pušu sazināšanos veica maršrutēšanas jeb saišu (*routes*) funkcijas Javascript valodā, šeit tas ir VueJS. Ziggy arī ļauj veikt saišu manipulācijas, kura tikai palielina tīmekļu vietnes drošību.

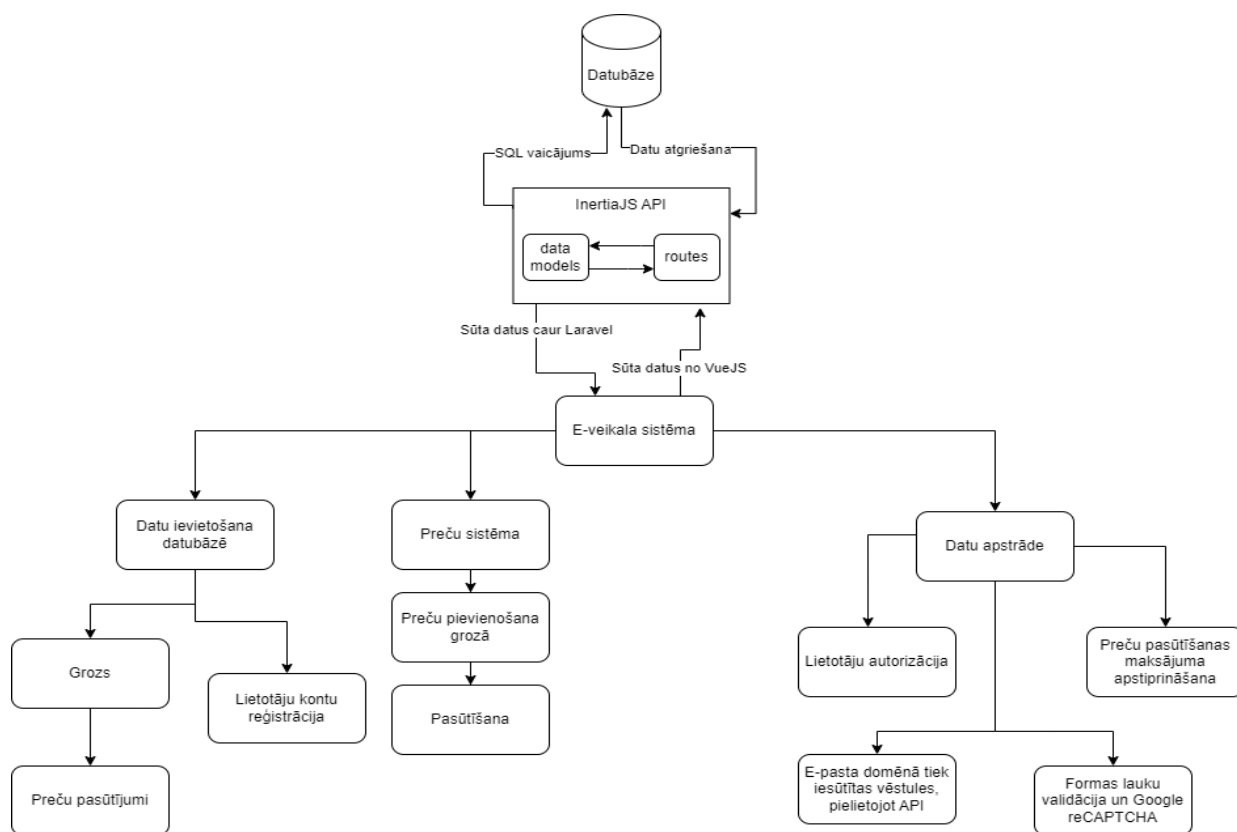
Lielākā daļu līdzekļu pamatojums lietošanā saistās ar skolas un prakses gūto pieredzi.

4. Sistēmas modelēšana un projektēšana

Sistēmas modelēšana un projektēšanas sadaļa tiek sastādīta, lai informētu lasītāju par sistēmas izstrādi caur shēmām un diagrammām. Sadaļa ir piemērota lasītājiem, kuri vēlas iepazīt sistēmu caur diagrammām. Dažkārt, diagrammas arī var palīdzēt iepazīt lasītājus, kuri nav saistīti ar lietotņu izstrādi vai ar IT.

4.1. Sistēmas struktūras modelis

Sistēmas struktūras modelis ir attiecināms pret sistēmas komponentu un to attiecību reprezentāciju. Tās mērķis ir sniegt palīdzību izstrādātājiem un sistēmas analītiķiem izprast, kā sistēma tiek izstrādāta un organizēta, tā sniedz pamatu sistēmas arhitektūrai, izveidei un analīzei. Modelis attēlo datu plūsmu un izveidoto savienojumu starp abām programmēšanas valodām caur API (skatīt 1. attēlu).

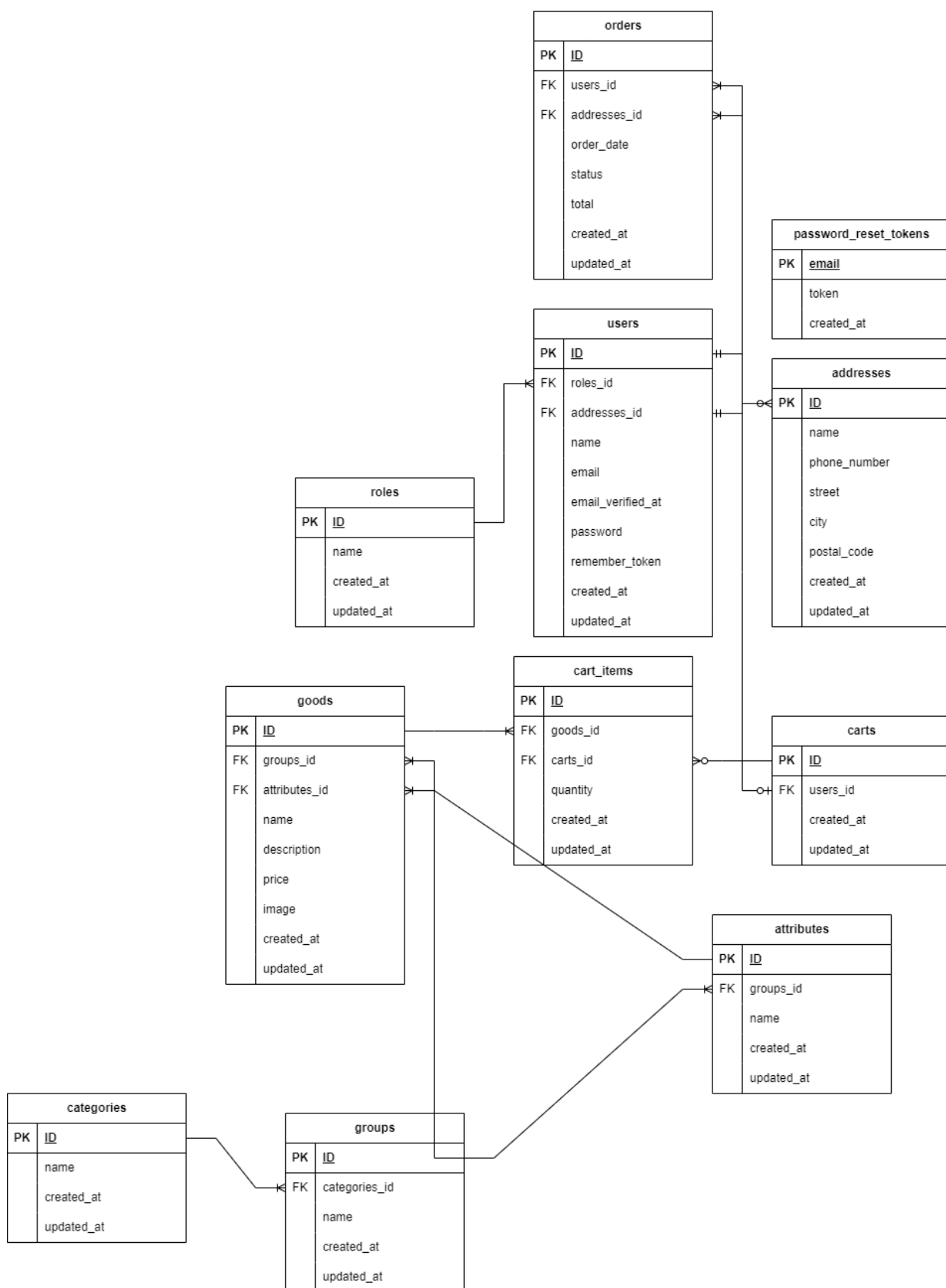


1. attēls. Datu plūsmas sistēmātiskais modelis.

4.1.1. ER diagramma

Entītiņu saistību diagrammas (ER diagrammas) ir datubāžu projektēšanas rīki, kuri tiek izmantoti, lai vizuāli attēlotu datubāzes struktūru un tajā esošo datu objektu (entītiņu) savstarpējās saites. Diagramma palīdz saprast datu organizāciju un to plūsmu.

Veidotā ER diagramma sastāv no divpadsmit tabulām. Lielāko daļu sastāv tabulas saistībā ar precēm, proti, preču kārtošana nepieciešams 3 atsevišķas tabulas. Tabulas saišu nolūkos, veidoti 2 daudzi-pret-daudziem saišu tipa tabulas, t.i. “order_details” un “cart_items”.



2. attēls. Datubāzes ER diagramma

4.2. Funkcionālais un dinamiskais sistēmas modelis

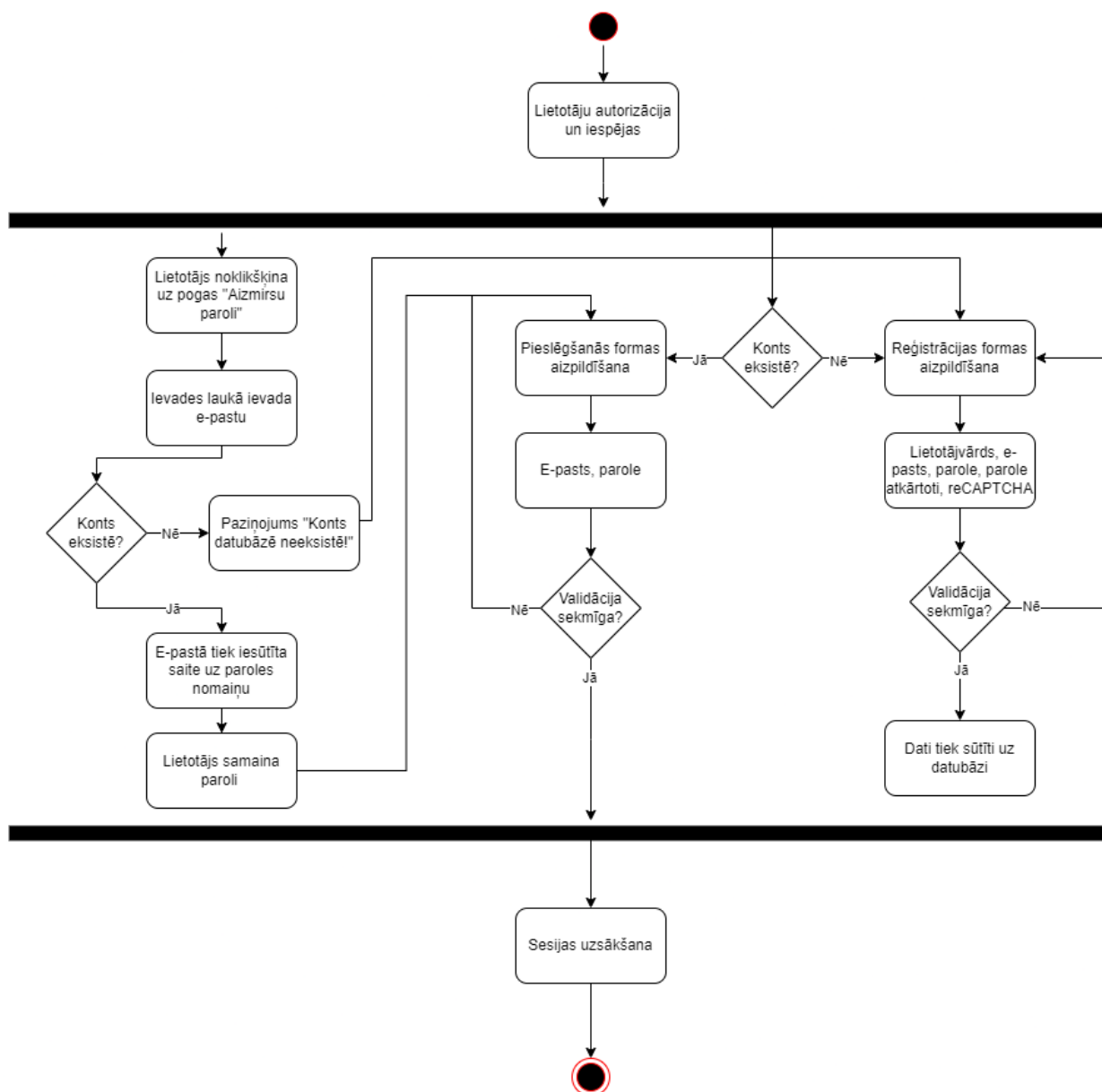
Funkcionālais un dinamiskais sistēmas modelis ir attiecināms uz sistēmas darbību un tās komponentu mijiedarbību. Šo modeļu mērķis ir palīdzēt izstrādātājiem un sistēmas analītiķiem izprast, kā sistēma darbojas un kādas ir tās komponentu attiecības. Funkcionālais modelis fokusējas uz sistēmas sniegtajām funkcijām un lietotāju mijiedarbību ar tām, izmantojot aktivitāšu un lietojumgadījumu diagrammas.

Dinamiskais modelis uzsver sistēmas uzvedību laika gaitā, attēlojot stāvokļu izmaiņas un komponentu mijiedarbību. Tas ietver sekvenču un stāvokļu diagrammas, kas ilustrē, kā sistēmas komponenti sadarbojas un kā tās reakcijas uz ārējiem notikumiem. Šie modeļi nodrošina detalizētu pārskatu par sistēmas darbību un palīdz identificēt potenciālas problēmas un uzlabojumus.

4.2.1. Aktivitāšu diagramma (Activity)

Aktivitāšu diagramma ir UML (Unified Modeling Language) diagrammas veids, kas parāda sistēmas darbības secību, izmantojot kontroli plūsmas konstrukcijas. Tas ietver dažādus elementus, piemēram, darbības, lēmumu mezglus, paralēlo izpildi un notikumu izraisīšanu. Aktivitāšu diagrammas bieži tiek izmantotas, lai vizualizētu darbplūsmas vai procesus sistēmā, jo tās spēj attēlot sarežģītu loģiku, kas ietver izvēles, ciklus un paralēlismu.

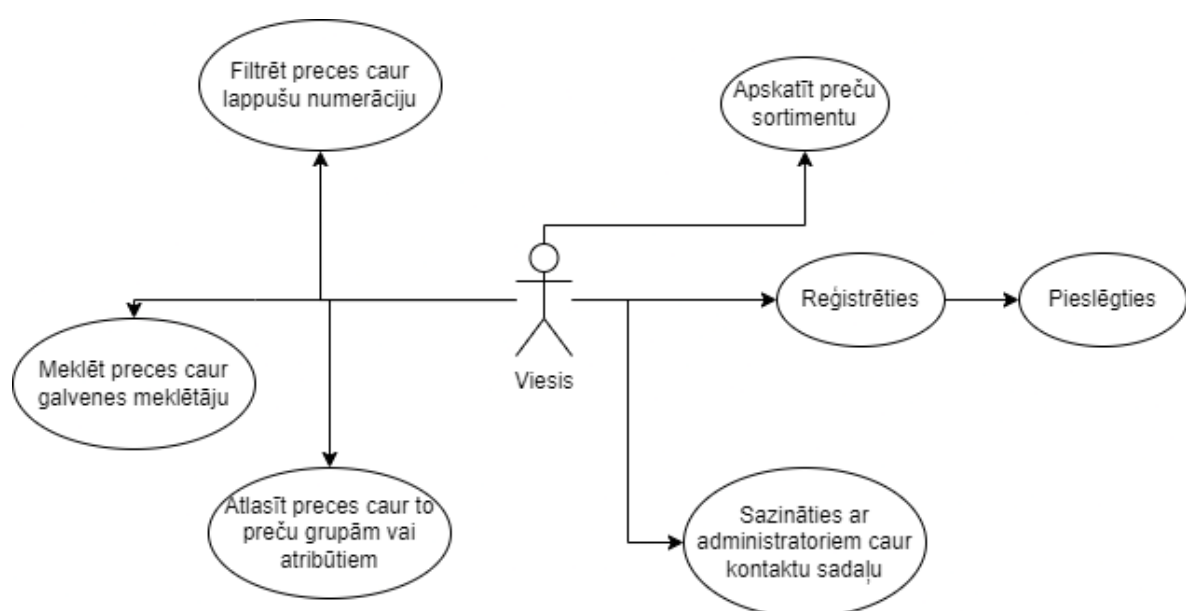
Lietotāju aktivitāšu diagramma attēlo lietotāja autorizācijas un reģistrācijas procesu sistēmā. Šāda veida diagramma sniedz vizuālu ceļvedi par to, kā lietotāji mijiedarbojas ar sistēmu, veicot noteiktas darbības (skatīt 3. attēlu).



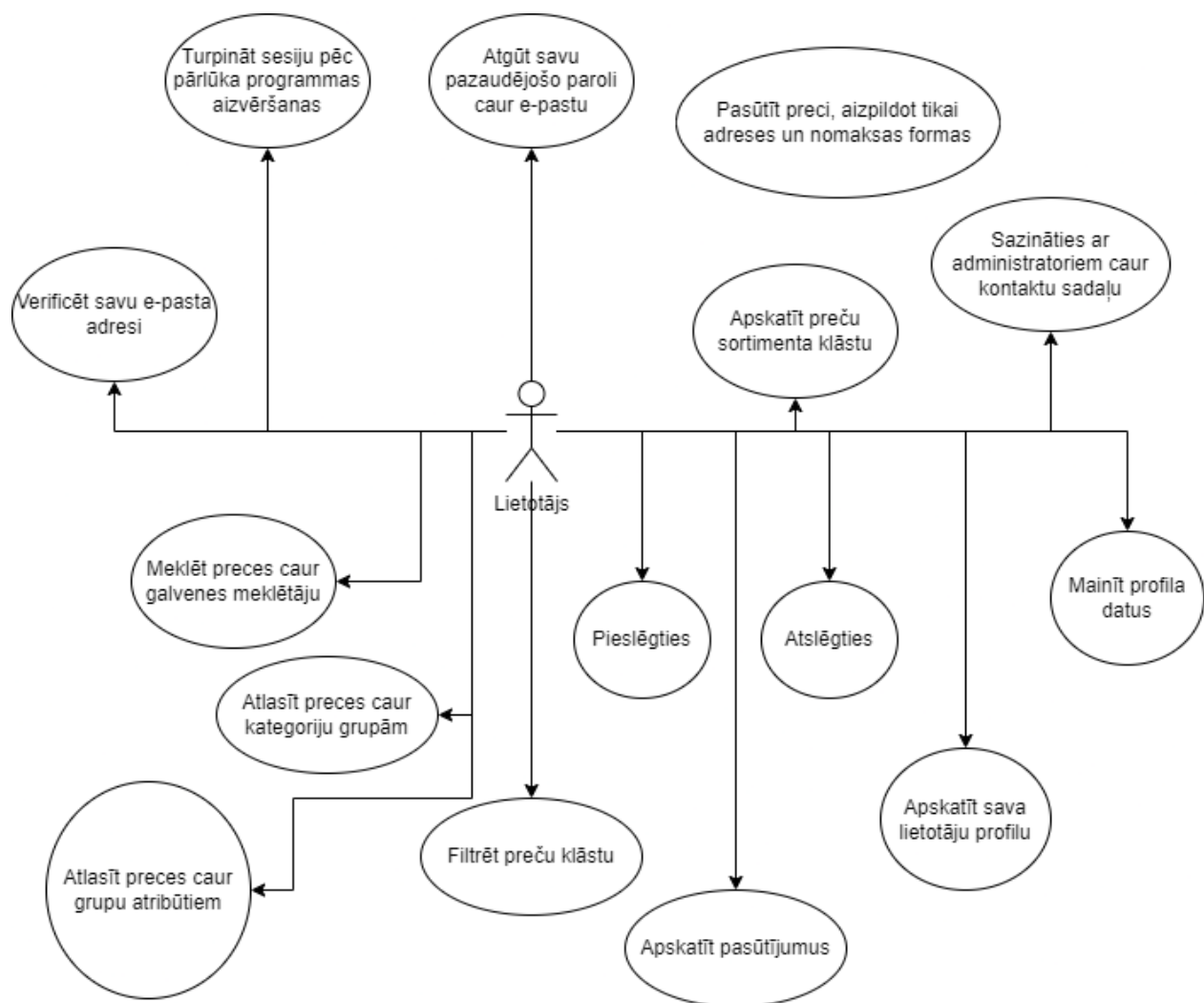
3. attēls. Lietotāju aktivitāšu diagramma.

4.2.2. Lietojumgadījumu diagramma (Use Case)

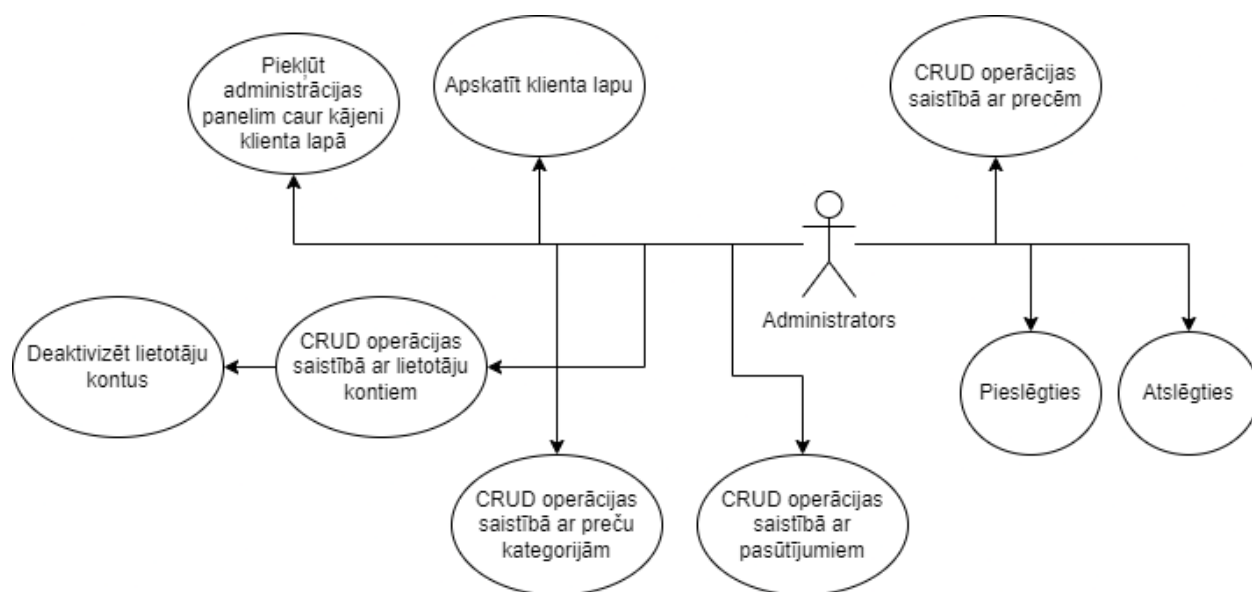
Lietojumgadījuma diagramma (*angļu valodā "use case diagram"*) ir UML (*Unified Modeling Language*) diagrammas veids, ko izmanto, lai aprakstītu sistēmas funkcionalitāti no lietotāja viedokļa. Tā parāda sistēmas dažādu lietotāju mijiedarbību ar sistēmas daļām, parasti atspoguļojot sistēmas galvenās funkcijas jeb "lietojumgadījumus" un to saistību ar dažādiem sistēmas lietotājiem. Tiek veidotas trīs lietojumgadījuma diagrammas, kas apraksta sistēmu no viesā, lietotāja un administratora skatpunktiem (skatīt 4., 5. un 6. attēlus).



4. attēls. Viesa lietojumgadījuma diagramma



5. attēls Lietotāja lietojumgadījuma diagramma.



6. attēls. Administratora lietojumgadījuma diagramma.

4.3. Sistēmas moduļu apraksts un algoritmu shēmas

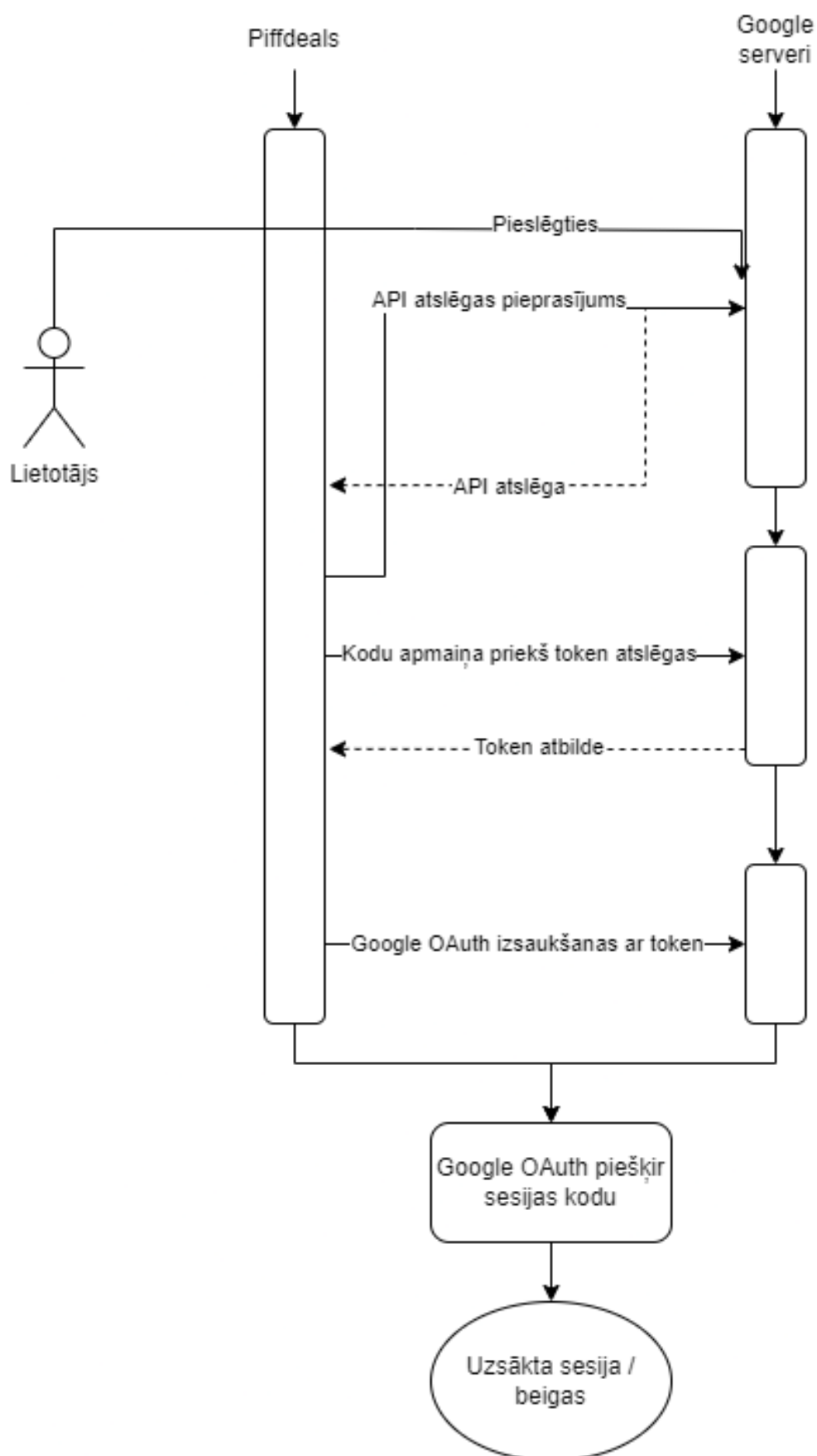
Šī nodaļa sniedz detalizētu sistēmas moduļu aprakstu un to algoritmu shēmas, koncentrējoties uz galvenajiem funkcionālajiem aspektiem. Sistēmas moduļu apraksts palīdz izprast katra komponenta lomu un funkcijas sistēmā, nodrošinot skaidru priekšstatu par to savstarpējo mijiedarbību un datu plūsmu.

Algoritmu shēmas piedāvā vizuālu pārskatu par galvenajiem procesiem sistēmā, īpaši pievēršot uzmanību Google/Facebook OAuth API konta reģistrācijas/pieslēgšanās procesiem, paroles atjaunošanas darbības principiem un token atslēgu veidošanai. Šīs shēmas ilustrē katra soļa secību un loģiku, kas veicina sistēmas drošību un efektivitāti.

4.3.1. Google OAuth pieslēgšanās algoritmu shēma

Diagramma ilustrē Google OAuth pieslēgšanās procesu Piffdeals tīmekļa vietnē. Pirmkārt, lietotājs uzsāk pieslēgšanās procesu, kas liek Piffdeals nosūtīt API atslēgas pieprasījumu Google serveriem. Pēc tam Google serveri pieprasa lietotājam pieslēgties un sniegt piekrišanu. Kad lietotājs apstiprina, Google ģenerē autorizācijas kodu, ko nosūta atpakaļ uz mājaslapu. Pēc tam Piffdeals apmaina šo kodu pret piekļuves tokena atslēgu, veicot vēl vienu pieprasījumu Google serveriem. Saņemot tokena atslēgu, Piffdeals izmanto to, lai izsauktu Google API un iegūtu lietotāja informāciju. Pēdējais solis ir sesijas kodu piešķiršana lietotājam, kas ļauj izmantot e-veikalu bez atkārtotas autentifikācijas. Konta neeksistējošie ieraksti datubāzē tiek izveidoti, paroles lauks netiek ievadīts. Šie soļi ir nepieciešami, lai nodrošinātu drošu un

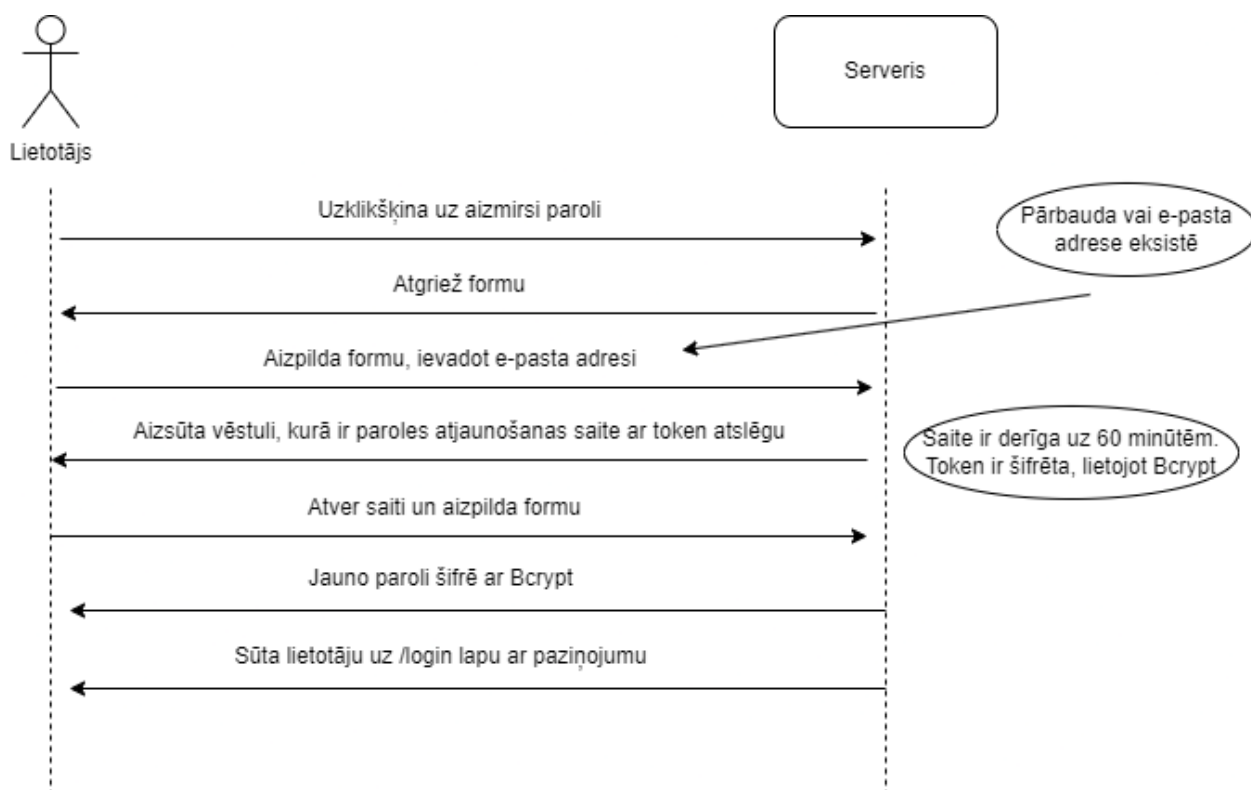
autorizētu piekļuvi lietotāja datiem, saglabājot datu konfidencialitāti un integritāti. Skatīt 7. attēlu.



7. attēls. Google OAuth pieslēgšanas algoritms

4.3.2. Paroles atjaunošanas algoritms

Paroles atjaunošanas algoritms ir būtiska funkcija, kas nodrošina lietotāju drošību un piekļuves iespējas situācijās, kad parole ir aizmirsta. Laravel ietvaros, šis algoritms piedāvā vienkāršu un drošu veidu, kā lietotāji var pieprasīt jaunu paroli, saņemot paroles atjaunošanas saiti savā e-pastā. Algoritms sākas ar lietotāja pieprasījumu paroles atjaunošanai, kas nosūta unikālu saiti uz e-pastu. Lietotājs klikšķina uz saites, aizpilda jaunas paroles formu, un sistēma atjauno paroli datubāzē. Šī funkcionalitāte ir nepieciešama, lai uzlabotu lietotāju pieredzi, samazinot piekļuves problēmas un palielinot konta drošību. Skatīt 8. attēlu.



8. attēls. Paroles atjaunošanas algoritms

5. Lietotāju ceļvedis

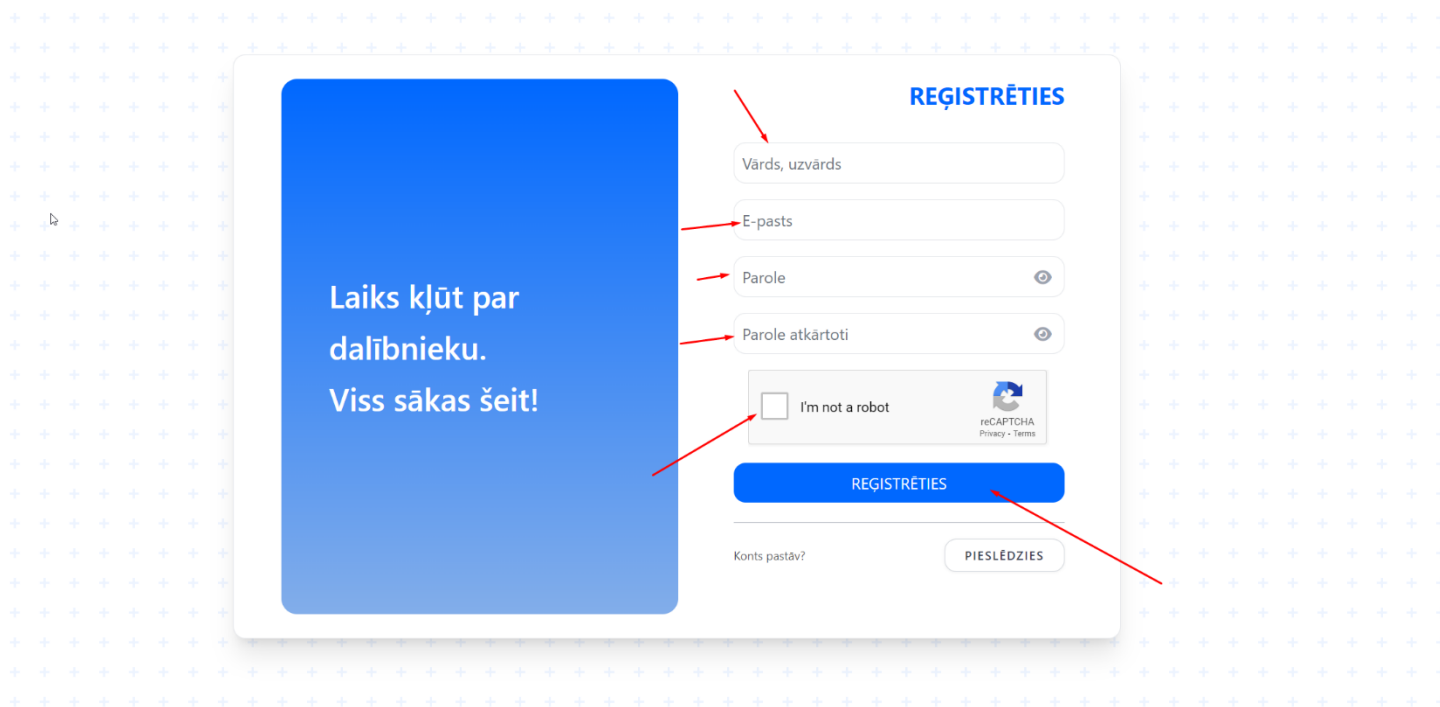
Lietotāja ceļvedis ir svarīgs rīks, lai palīdzētu jauniem un esošiem klientiem efektīvi izmantot e-veikala piedāvātās iespējas un funkcijas. Šāds ceļvedis nodrošina instrukcijas un padomus par to, kā veikt pirkumus, izmantot dažādas meklēšanas un filtrēšanas opcijas, pārvaldīt lietotāja kontu, saprast piegādes un atgriešanas politiku, kā arī izmantojot klientu atbalsta pakalpojumus.

5.1. Reģistrācija e-veikalā kā viesis

Reģistrācija e-veikalā nav obligāta, taču veicot preču pasūtījumus un veidot savu iepirkumu grozu, autorizācija ir nepieciešama. Reģistrācija sistēmā sniegs iespēju klientiem veikt izmaiņas profila sadaļā, pievienot preces grozā un veikt pasūtīšanu. Reģistrāciju var veikt uzklikšķinot uz pogas “Pieslēgties”, pēc tam uz pogas “Reģistrējies”. Skatīt 9. un 10. attēlus.



9. attēls. Bulta uz pogas “Pieslēgties”



10. attēls. Reģistrācijas forma

5.2. Pieslēgšanās e-veikalā, lietojot Google OAuth

Lietojot Google OAuth, lietotājam nav jāuztraucas par korektu e-pastu un paroles datu ievadi. Uzklīkšķinot uz pogas “Google”, lietotājs tiek novirzīts uz Google lapu, kurā jāizvēlas ar, kuru kontu pieslēgties. Ja Google e-pasts neeksistē datu bāzē, tad tā tiks izveidota. Skatīt 11. attēlu.

The screenshot shows a login interface with the title "PIESLĒGTIES". It includes input fields for "E-pasts" and "Parole" (password), a checkbox for "Atcerēties mani", and a blue "PIESLĒGTIES" button. Below this is a "VAI" separator and a Google login button with the Google logo. A red arrow points to the Google button. To the right of the login form is a large blue box with white text: "Mūsu biznesa mērķis? Digitalizēt iepirkšanos tiešsaistē." At the bottom left, there are links for "Aizmirsti paroli?" and "Nav konta?" with a "REGISTRĒJIES" button.

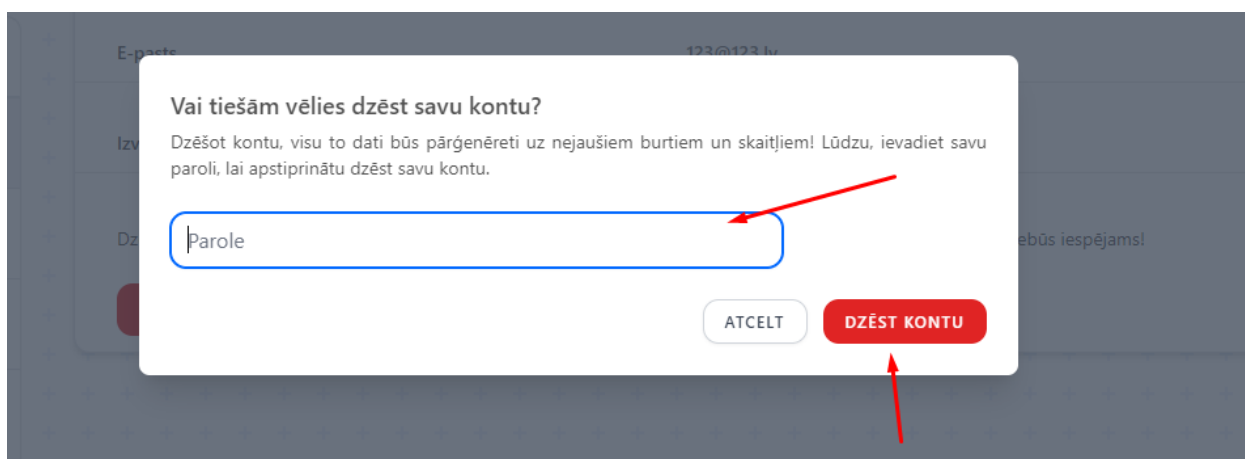
11. attēls. Pieslēgšanās, lietojot Google OAuth

5.3. Konta dzēšana profilu sadaļā

Konta dzēšanu var veikt jebkurš autorizēts lietotājs. To ir iespējams izdarīt profilu sadaļā, uzspiežot uz pogas “Dzēst kontu” un apstiprināt darbību, ievadot konta paroli. Konta dzēšana, klienta ietvaros, anonimizē klienta datus, proti, konta vārds, uzvārds un e-pasts, parole u.c. tiek anonimizēti uz nejaušiem burtiem un skaitļiem. Dzēšot klienta kontu no datubāzes ir iespējams tikai pēc pieprasījuma, sazinoties ar administrāciju kontakta lapā. Skatīt 12. un 13. attēlus.

The screenshot shows a confirmation page for deleting an account. It displays the user's details: "Vārds, uzvārds" (123), "E-pasts" (123@123.lv), and "Izveidots" (pirms 2 stundām). Below this is a warning message: "Dzēšot savu lietotāja kontu, visi dati datubāzē tiks anonimizēti uz nejaušiem burtiem un skaitļiem. Kontu atgūt nebūs iespējams!". At the bottom is a red button labeled "DZĒST KONTU" with a red arrow pointing to it.

12. attēls. Poga “Dzēst kontu”



13. attēls. Konta dzēšanas apstiprināšana

5.4. Paroles atjaunināšana caur vēstuli

Paroles nozaudēšanas gadījumā, lietotājs ir tiesīgs pieprasīt paroles atjaunošanu, pieprasot to caur e-pastu. Ievadot derīgu e-pasta adresi, lietotājam uz e-pastu atnāks vēstule, kurā ir “Atiestatīt paroli” poga uz, kuras uzklikšķinot, atver privātu lapu, proti, lapai tiek uzģenerēta token tipa atslēga un ievades laukā tiek ielādēta pieprasītā lietotāja e-pasts. Tas nodrošina drošību pret ļaunprātīgu funkcijas izmantošanas, t.i. citu kontiem nevar samainīt paroli un token atslēgas termiņš beidzas pēc 60 minūtēm. Skatīt 13., 14, 15. un 16. attēlus.

PIESLĒGTIES

E-pasts

Parole

☐ Atcerēties mani

PIESLĒGTIES

— VAI —

Google

[Aizmirsi paroli?](#)

Nav konta? **REĢISTRĒJIES**

Mūsu biznesa mērķis? Digitalizēt iepirkšanos tiešsaistē.

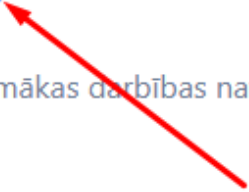
14. attēls. "Aizmirsi paroli" poga pieslēgties logā

Piffdeals

Sveiki!

Jūs saņēmāt šo e-pastu, jo mēs saņēmām paroles atiestatīšanas pieprasījumu jūsu kontam.

Atiestatīt Paroli



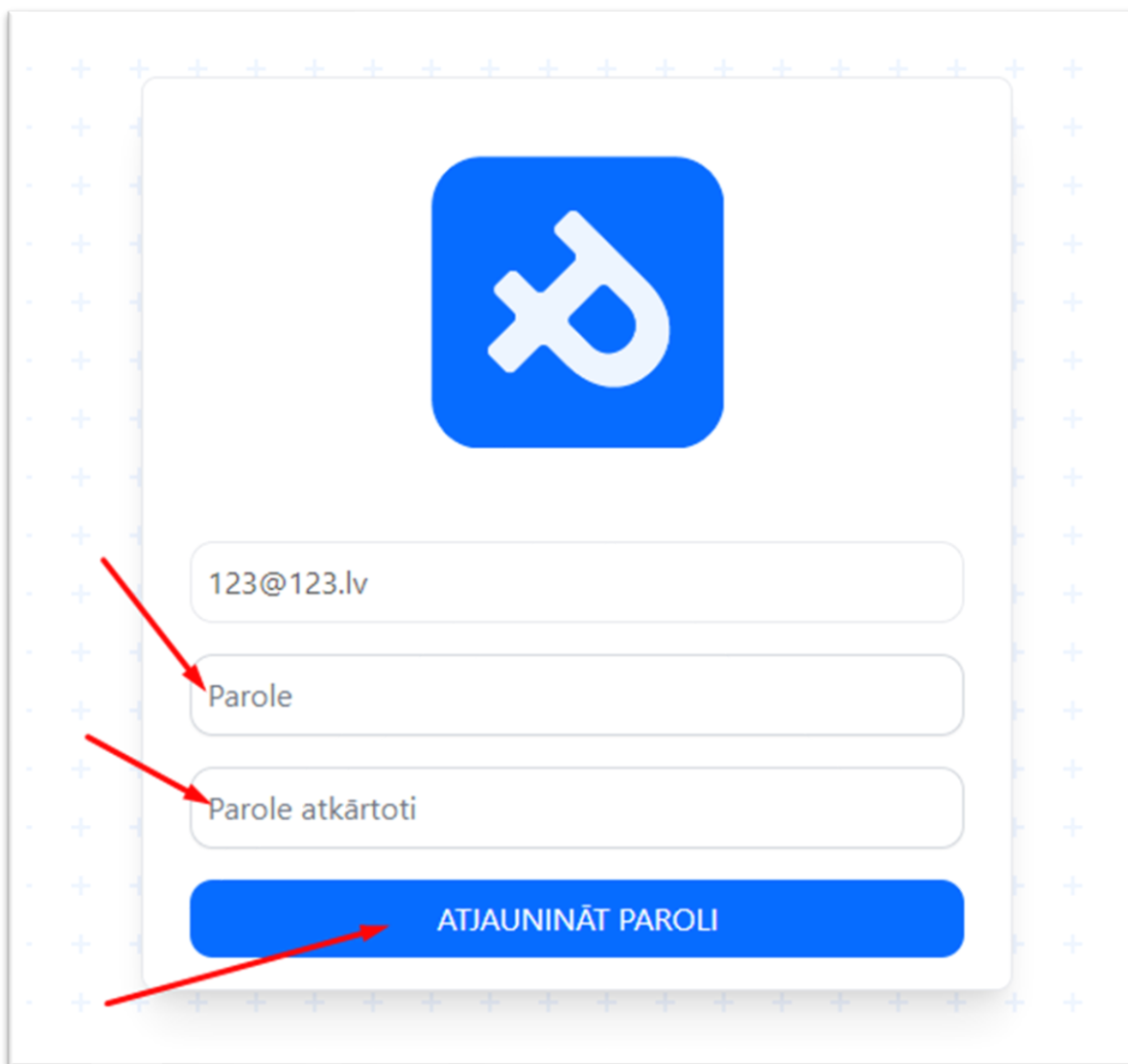
Ja jūs nepieprasījāt paroles atiestatīšanu, turpmākas darbības nav nepieciešamas.

Ar cieņu!
Piffdeals

Ja rodas problēmas klikšķinot uz "Atiestatīt Paroli" pogas, tad droši pārkopē šo doto saiti savā pārlūkprogrammā: <http://localhost:8000/reset-password/db8bf09d5b9037b432d23413482c6ce3513435727fb525542118ffb31ad84fa?email=123%40123.lv>

© 2024 Piffdeals. Visas tiesības aizsargātas.

15. attēls. Paroles atjaunošana vēstule e-pastā



16. attēls. Paroles atjaunošanas forma

5.5. Jaunu preču izveidošana

Administrators ir tiesīgs pievienot preces administrācijas panelī. Poga “Pievienot preci” atver modāla loga, kurā nepieciešams ievadīt svarīgāko informāciju, piemēram, preces noliktavas SKU. Pēc izvēles, preci var pievienot zem kategoriju grupas vai grupu atribūta. Izveidojot preci, administrators tiek novirzīts uz preču rediģēšanas lapu, kurā nepieciešams ievadīt preču datus. Skatīt 17., 18., 19. attēlus.



17. attēls. Preču pievienošanas poga

PIEVIENTOT JAUNU PRECI

SKU

Izvēlies atribūtu

Izvēlies grupu

ATCELT

PIEVIENTOT

18. attēls. Preču pievienošanas modāla logs

REDIGĒT PRECI

ATRAKAT

SAGLABĀT PRECI

PRECES INFORMĀCIJA

Statuss

Deaktivizēts

Cena

€ 0

Preču skaita nošķiršana

0

Nosaukums

Nenosaukta prece

PRECES KĀRTOŠANA KATEGORIJĀ

Atribūts

Grupa

BĒST PRECI

19. attēls. Preču rediģēšanas lapa

5.6. Profila informācijas atjaunināšana

Autorizēts lietotājs ir tiesīgs mainīt savu e-pasta adresi vai vārdu, uzvārdu jebkurā laikā, apmeklējot profila sadaļu. Skatīt 20. attēlu.

Profilā informācija
Šeit Jūs varat atjaunināt sava profila vārdu, uzvārdu un e-pasta adresi.

Vārds, uzvārds

E-pasts

SAGLABĀT

20. attēls. Profila informācijas forma

5.7. Paroles maiņa profila sadaļā

Autorizēts lietotājs ir tiesīgs mainīt paroli, aizpildot paroles maiņas formu. Skatīt 21. attēlu.

Atjaunināt paroli
Izvairieties no vienkāršu vai viegli uzminamu parolu lietošanas, piemēram, dzimšanas datumi vai vārdi.

Pašreizējā parole

Jaunā parole

Jaunā parole atkārtoti

SAGLABĀT

21. attēls. Paroles maiņas forma

5.8. Kontakta veidlapas aizpildīšana

E-veikala klienti gan autorizēti, gan e-veikala viesi ir tiesīgi sazināties ar e-veikala administrāciju sadaļā “Sazinies ar mums”, aizpildot formu, ievadot derīgu e-pasta adresi. Skatīt 22. attēlu.

Sazinies ar mums

Saskārāties ar tehniskām grūtībām, lietojot mūsu e-veikalu? Vēlaties dalīties ar atsauksmēm par mūsu tīmekļa vietnes funkcijām? Mēs esam šeit, lai jums palīdzētu!

E-pasts: *

piemers@epasts.lv

Temats: *

Kā mēs varam palīdzēt?

Jūsu ziņojums: *

Paskaidro situāciju šeit



I'm not a robot



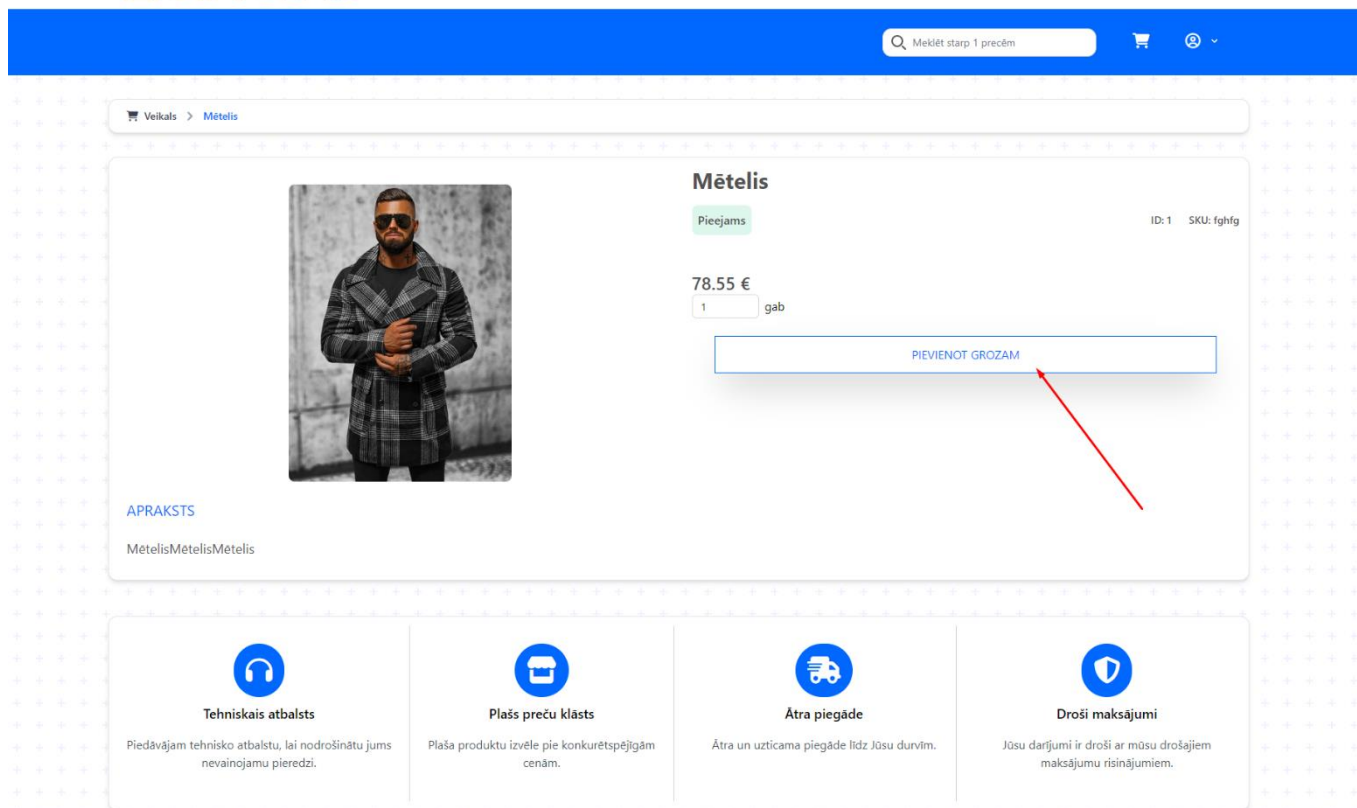
Sūtīt ziņu

22. attēls. Kontakta saziņas veidlapa

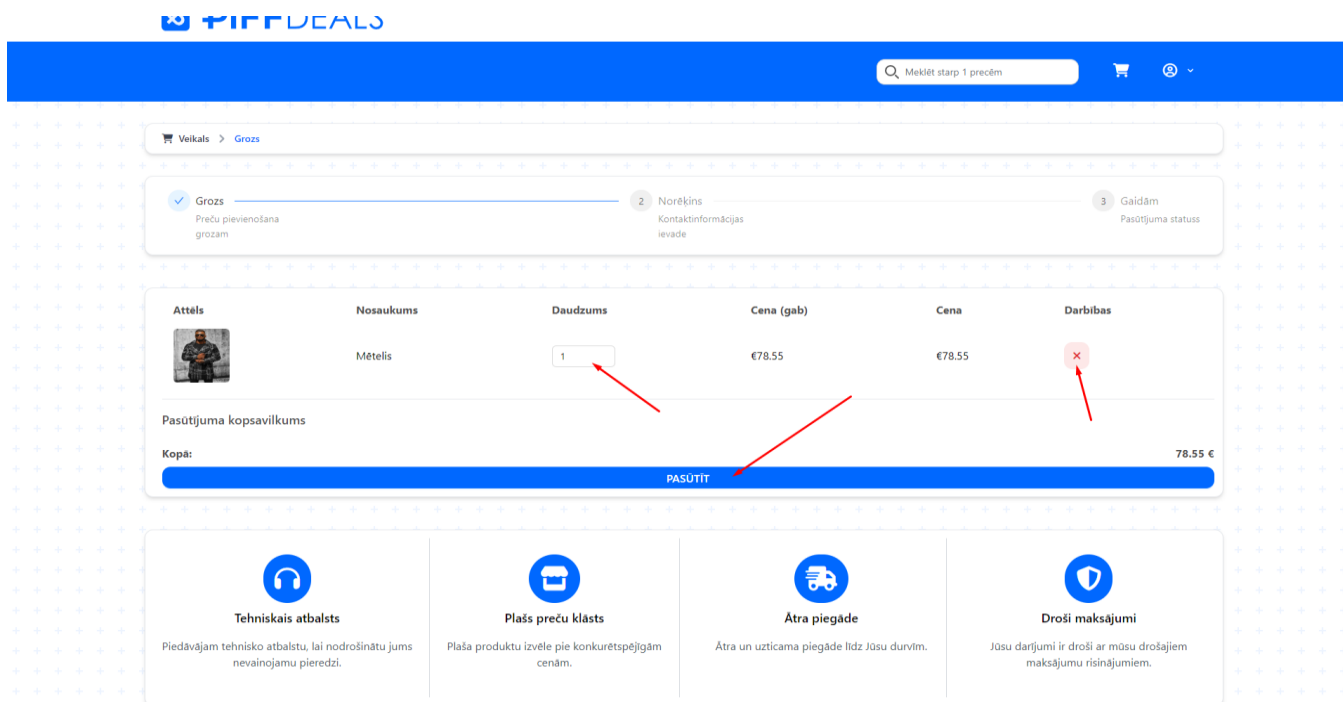
5.9. Preču ievietošana grozā

E-veikala reģistrēti, e-pastu verificēti, lietotāji ir tiesīgi pievienot preces grozam, kā arī izņemt preces no groza un veikt preču pasūtīšanu. Preču pievienošana grozā nav sarežģīts process. Skatīt 23. un 24. attēlus.

PIFFDEALS



23. attēls. Preču pievienošana grozam



24. attēls. Groza preču darbības

6. Testēšanas dokumentācija

Kvalifikācijas eksāmena praktiskās daļas dokumentācijā sadaļa "Testēšanas dokumentācija" ir paredzēta, lai aprakstītu detalizētu e-veikala sistēmas testēšanas procesu. Tā sniedz izpratni par izvēlētajām testēšanas metodēm, izmantotajiem rīkiem, kā arī pamatojumus par to izvēli. Tālāk ir apkopoti konkrēti testpiemēri un metodēm atbilstoša testēšanas gaita, kuras rezultāti tiek fiksēti testēšanas žurnālā.

6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums

Šajā sadaļā tiek aprakstītas specifiskas testēšanas metodes, kas tika izmantotas sistēmas drošības, funkcionalitātes, lietotāja pieredzes un atbilstības prasībām pārbaudei. Testēšana ietver gan automatizētos, gan manuālos testus, kas nodrošina sistēmas kvalitātes kontroli visā tās izstrādes un implementācijas ciklā. Izvēlētie rīki, piemēram, PHPUnit un Laravel Dusk, ir detalizēti aprakstīti ar norādi uz to efektivitāti un integrācijas iespējām projektā.

Izstrādātājs ir izvēlējis veikt manuālu testēšanu, jo sistēma nav pietiekami apjomīga, lai veiktu automatizētus testus, tostarp rakstīt testēšanas skriptus un lietot citas papildu programmatūras. Testēšanas metode būs *Black Box* – testētājs pārbauda programmatūras funkcionalitāti bez zināšanām par to, kā kods darbojas. Testētājs koncentrējas uz programmatūras saskarnes un izvades pārbaudi, izmantojot ievades datus un pārbaudot izvadi, neizpētot tās iekšējo kodu vai struktūru. Galvenais mērķis ir noteikt, vai programmatūra atbilst specifikācijām un izpilda paredzētās funkcijas pareizi. Netiks lietota *White Box* – pretēji black box metodei, prasa, lai testētājam būtu zināšanas par programmatūras iekšējo kodu, arhitektūru un darbības principiem. Šī metode ļauj izpētīt sistēmas iekšējos resursus, izpildot kodu, pārbaudot datu plūsmu un citus tehnikas aspektus, lai identificētu iespējamās kļūdas. *White box* testēšanā testētāji var efektīvi atrast loģikas kļūdas, drošības problēmas un citas kodējuma problēmas, jo pasūtītājs neatļauj publicēt programmatūras svarīgāko koda daļu, tas nozīmē, testētājam būtu jāapraksta katra funkcionalitāte un tās darbības gaita.

6.2. Alternatīvās testēšanas metodes un rīki

Šī sadaļa apraksta dažādus alternatīvos testēšanas rīkus un metodes, kas var tikt izmantoti e-veikala tīmekļa vietnes izstrādes un testēšanas procesā. Alternatīvās metodes un rīki ir svarīgi, jo tie nodrošina papildu testēšanas iespējas, kas var palīdzēt identificēt problēmas, kas citādi varētu palikt nepamanītas. Šīs metodes un rīki ļauj izstrādātājiem izvēlēties piemērotāko pieeju konkrētajam projekta scenārijam un nodrošina plašāku pārbaudes segumu.

Pastāv vairākas alternatīvas testēšanas metodes un tām ir izšķirīga nozīme programmatūras izstrādes procesā. Tas ir nepieciešams, lai nodrošinātu gala produkta kvalitāti.

Vienības testēšana (*Unit Testing*) – metode ietver koda mazākās daļas testēšanu, lai pārbaudītu, vai tās atsevišķi darbojas pareizi. Parasti, vienības testēšana, ir automatizēts process, kurā var ātri identificēt problēmas komponentos.

Integrācijas testēšana (*Integration Testing*) - pēc vienības testēšanas veikšanas seko integrācijas testēšana, kur tiek pārbaudīts, kā dažādas sistēmas daļas strādā kopā. Šī metode koncentrējas uz datu plūsmu un savienojumiem starp moduļiem vai sistēmām.

Slodzes testēšana (*Load Testing*) - slodzes testēšana tiek veikta, lai novērtētu, kā sistēma darbojas, kad tai tiek uzlikta liela slodze, piemēram, liels lietotāju skaits, liela datu apstrāde utt. Šī testēšana palīdz noteikt sistēmas veiktspējas robežas.

Selenium – automatizēts pārlūkprogrammas testēšanas rīks, ideāli piemērots dinamisku tīmekļa vietņu funkcionalitātes un saskarnes testēšanai.

JUnit – populārs Java testēšanas rīks, kas ļauj veikt vienības testēšanu. Programmatūru parasti izmanto, lai testētu back-end loģiku.

Postman - API rīks testēšanai, kas ļauj veikt pieprasījumus uz serveri un pārbaudīt atbildes. Postman ir noderīgs API integrācijas pārbaudei, nodrošinot viegli lietojamu saskarni testēšanas scenāriju izstrādei.

6.3. Testpiemēru kopa

Apkopojot praktiskos testpiemērus, tiek izmantota testu kopas izveidošanas stratēģija, kas ietver dažādu scenāriju simulēšanu — gan standarta darbību veikšanu, gan iespējamo kļūdu un pārtraukumu radīšanu. Katrs testpiemērs ir aprakstīts, norādot ievaddatus, gaidāmo uzvedību un vēlamo izvadi, lai nodrošinātu visaptverošu pārskatu par sistēmas atbilstību izvirzītajām prasībām. Skatīt testpiemēru kopu.

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra apraksts	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra sagaidāmais rezultāts	Prasības ID
Black Box							
TP.KL.01	Aktīvu kategoriju ielāde	Kategorijai ir jābūt statusam „Aktīvs” un jābūt savienotai ar vienu vai vairākām preču grupām	Kategoriju uzrāde klientu pusē galvenes joslā	Administratoram jāizveido kategorija un jāsavieno ar preču grupu	Mājaslapas ielāde klienta pusē	Tiek ielādētas kategorijas galvenes joslā	P.02
TP.KL.02	Preču skaits meklēšanas joslā	Klientam vai administratoram jāapmeklē klientu puse. Administratoram jāizveido preces	Aktīvo preču skaits tiek ielādēts meklēšanas joslā	Administratoram ir jāizveido preces ar statusu „Aktīvs”	Administratoram jāizveido preces sadaļā „Preces”	Preču skaits mainās ik reizi preču statuss mainās uz “Aktīvs”	P.03
TP.KL.03	Preču filtrēšana caur meklēšanas joslu	Administratoram jāizveido preces. Jāievada preču atslēgasvārdi meklēšanas joslā	Meklēšanas josla izvada meklēšanas logu ar rezultātu	Jāievada atslēgasvārdi meklēšanas joslā	Ar tastatūras palīdzību, ievadīt atslēgasvārdus	Tiek veikta preču filtrēšana	P.06
TP.KL.04	Kontaktu veidlapas aizpildīšana	Lietotājam jāapmeklē sadaļa “Sazinies ar mums” un jāaizpilda forma	Lietotājs var sazināties ar administrāciju, aizpildot formu	Aizpildīt formu, norādot derīgu e-pasta adresi	Ar tastatūras palīdzību, aizpildīt formu	E-pasta ziņa tiek nosūtīta uz e-pastu: support@piffdeals.lv	P.01
TP.KL.P.05	Profilu sadaļas piekļuve	Jābūt autorizētam un jāuzspiež uz pogas „Profils”	Autorizēts lietotājs ir tiesīgs apmeklēt sadaļu “Profils”	Lietotājm jāautorizējas un jāuzklikšķina uz pogas “Profils”	Lietotājs pieslēdzas sistēmai caur /login	Lietotājs piekļūst sadaļai “Profils”	P.11
TP.KL.P.06	Konta e-pasta vai vārda, uzvārda maiņa profila sadaļā	Jābūt autorizētam un jāuzspiež uz pogas „Profils”	Autorizēts lietotājs ir tiesīgs veikt e-pasta vai vārda, uzvārda maiņu	Lietotājam jāapmeklē sadaļa „Profils” un jāaizpilda rediģēšanas forma, jānospiež poga „Saglabāt”	Lietotājam jāapmeklē profila sadaļa	Lietotājs ir sekmīgi mainījis konta e-pastu vai vārda, uzvārda datus	P.30

TP.KL.P.07	Konta paroles maiņa profila sadaļā	Jābūt autorizētam un jāuzspiež uz pogas „Profils”	Autorizēts lietotājs ir tiesīgs mainīt konta paroli	Lietotājam jāapmeklē sadaļa „Profils” un jāaizpilda paroles maiņas forma, jānospiež poga „Saglabāt”	Lietotājam jāapmeklē profila sadaļa	Lietotājs ir sekmīgi mainījis konta paroli	P.33
TP.KL.P.08	Konta dzēšana	Jābūt autorizētam un jāuzspiež uz pogas „Profils”	Autorizēts lietotājs ir tiesīgs dzēst kontu sistēmā	Lietotājam jāapmeklē sadaļa „Profils” un jāuzspiež uz pogas „Dzēst kontu”, jāveic apstiprinājums, ievadot konta paroli un jānospiež „Dzēst kontu” poga	Lietotājam jāapmeklē profila sadaļa	Lietotājs ir sekmīgi dzēsis jeb anonimizējis savus profila datus. Sesija tiek pārtraukta	P.31
TP.KL.P.09	Paroles atiestatīšana /login lapā	Neautorizētam lietotājam jānospiež „Aizmirsi paroli?” poga	E-veikala viesis ir tiesīgs veikt paroles atjaunošanu eksistējošam kontam	Lietotājam ir jāaizpilda „aizmirsi paroli” forma, jāsaņem vēstule e-pastā, jāuzklikšķina uz atiestatīšanas pogas un jāaizpilda paroles maiņu dati	Lietotājam nav jābūt autorizētam un jāapmeklē /login lapā	Lietotāja konta parole tiek mainīta, neuzsākot sesiju	P.25
TP.KL.P.10	Konta e-pasta verifikācija	Lietotājam jāreģistrē konts	Verificēt kontu caur e-pasta vēstules	Lietotājam reģistrācijas lapā, jāaizpilda forma un jānospiež poga e-pastas vēstulē	Lietotājam jāreģistrējas	Lietotāja konts ir veiksmīgi verificēts	P.28
TP.KL.PAT.11	Preču atlase no kategoriju grupas	Administratoram jāizveido preces un jāveic kārtošana pa grupām. Klientam jāuzspiež vienā no kategorijas grupām	Jaunas lapas ielāde ar atlasītām precēm pēc to kategoriju grupas	Klientam, galvenes joslā, jāuzliek peles kursoru uz kategorijas, kas atlasītu kategorijas grupas	Administratoram jāizveido preces un jāveic kārtošana	Tiek uzģenerēta jauna lapa, kurā ir atlasītas preces no to grupas	P.14
TP.KL.PAT.12	Preču atlase no grupas atribūta	Administratoram jāizveido preces un jāveic kārtošana pa grupu atribūtiem. Klientam jāuzspiež vienā no grupas atribūtiem.	Jaunas lapas ielāde ar atlasītām precēm pēc to grupu atribūta	Klientam, galvenes joslā, jāuzliek peles kursoru uz kategorijas, kas atlasītu kategorijas grupas	Administratoram jāizveido preces un jāveic kārtošana	Tiek uzģenerēta jauna lapa, kurā ir atlasītas preces no to grupu atribūta	P.13

TP.KL.PR.13	Preču pievienošana grozā	Reģistrētam, verificētam lietotājam ir jāuzspiež uz pogas “Pievienot grozam”	Pievienot preces grozā ar datubāzes palīdzību	Lietotājam ir jāreģistrē konts un jāverificē, pēc tam preces logā uzspiež uz pogas “Pievienot grozā”	Administratoram jāizveido aktīvas preces un lietotājam ir jāuzspiež uz pogas “Pievienot grozam”	Prece tiek veiksmīgi pievienota grozā	P.16
TP.KL.PR.14	Groza preču manipulācijas	Reģistrētam, verificētam lietotājam ir jāpievieno preces grozā un pēc izvēles, ir tiesīgs mainīt preču skaitu vai izdzēst no groza	Lietotājs ir tiesīgs veikt preču groza manipulācijas jeb veikt dažādas darbības	Lietotājam ir jāpievieno preces grozā un jāapmeklē lapa /cart, pēc izvēles ir tiesīgs spiest uz “X” pogas vai mainīt preču skaitu	Lietotājam jāpievieno preces grozā	Lietotājs ir veiksmīgi veicis groza preču datu manipulācijas	P.18
TP.KL.PR.15	Jaunu adrešu izveide	Pēc grozu preču manipulēšanas, spiest uz pogas „PASŪTĪT”, kurā lietotājs /checkout lapā aizpilda formu	Lietotājs pasūtīšanas procesā aizpilda kontaktinformācijas formu	Lietotājam grozā jāpievieno vismaz viena prece un jāuzspiež uz pogas “Pasūtīt”	Lietotājam jāpievieno preces grozā un jāuzspiež uz pogas “PASŪTĪT” un lapā /checkout jāaizpilda kontaktinformācija	Lietotājs ir veiksmīgi izveidojis pasūtījumu	P.19
TP.KL.PR.16	Jaunu pasūtījumu izveide	Pēc grozu preču manipulēšanas, spiest uz pogas „PASŪTĪT”, kurā lietotājs /checkout lapā aizpilda formu, un uzspiež uz pogas “PASŪTĪT”	Lietotājs pasūtīšanas procesā aizpilda kontaktinformācijas formu un nospiež pogu “PASŪTĪT”	Lietotājam grozā jāpievieno vismaz viena prece un jāuzspiež uz pogas “PASŪTĪT”, aizpilda kontaktinformācijas formu un uzspiež uz pogas „PASŪTĪT”	Lietotājam jāpievieno preces grozā un jāuzspiež uz pogas “PASŪTĪT” un lapā /checkout jāaizpilda kontaktinformācija	Lietotājs ir veiksmīgi izveidojis pasūtījumu	P.20
TP.AP.17	Jaunu preču izveidošana	Administratoram ir jābūt autorizētam apmeklēt lapu /admin-goods un jāuzspiež uz pogas “Pievienot preci”	Administrators izveido jaunu preci	Administratoram jāuzspiež uz pogas „Pievienot preci” un jāaizpilda forma	Jāuzspiež uz pogas „Pievienot preci”	Administrators ir veiksmīgi izveidojis preci, taču nav to rediģējis	P.21
TP.AP.18	Eksistējošā preču rediģēšana	Administratoram ir jāizveido jauna prece lapā /admin-goods	Administrators ir tiesīgs rediģēt eksistējošu preci tabulā	Administrators uzspiež uz preču rediģēšanas pogas, kas novirza lietotāju uz preču rediģēšanas formas lapu	Tikai eksistējoša prece datubāzē ļauj to rediģēt	Administrators nospiežot pogu „Saglabāt” atjaunina preces datus datubāzē	P.32

TP.AP.19	Jaunu klienta kontu izveide	Administratoram jāapmeklē lapa /admin-users un jāuzspiež uz pogas "Izveidot kontu"	Administrators izveido jaunu klienta kontu	Administrators uzspiež uz pogas „Izveidot kontu”, aizpilda modāla logā konta izveidošanas formu un uzspiež uz pogas „Izveidot”	Administratoram jāapmeklē /admin-users lapa	Administrators ir veiksmīgi izveidojis klienta kontu	P.39
TP.AP.20	Paroles atiestatīšanas vēstule	Administratoram jāapmeklē /admin-users lapa un uz eksistējoša klienta konta, tabulā jāuzspiež uz paroles atiestatīšanas pogas	Administrators ir tiesīgs aizsūtīt paroles atiestatīšanas vēstuli uz eksistējoša klienta konta	Administratoram jāuzspiež uz paroles atiestatīšanas pogas un jāapstiprina sūtījums	Jāuzspiež uz pogas, pēc tam jāapstiprina sūtījums	Klienta kontam ir pienācis paroles atiestatīšanas vēstule	P.41
TP.AP.21	Konta dzēšana no datubāzes	Administratoram jāapmeklē /admin-users lapa un uz eksistējoša klienta konta, tabulā jāuzspiež uz apskates pogas	Administrators ir tiesīgs apskatīt detalizētu konta datus un pēc izvēles dzēst kontu no datubāzes	Administratoram jāuzspiež uz pogas "Dzēst kontu" un jāveic apstiprinājums	Jāuzspiež uz pogas, pēc tam jāapstiprina	Klientu konts ir izdzēsts no datubāzes	P.40

Testpiemēru kopa

6.4. Testēšanas žurnāls

Testēšanas žurnāls satur detalizētu informāciju par veikto testēšanu, tostarp datumu, testēšanas sesiju rezultāti, konstatētās problēmas un to novēršanas gaita. Dokumentējot visu testēšanas procesu, tiek nodrošināta caurskatāmība un atskaitīšanas iespējas projekta komandai un interesentiem. Testēšanas žurnāls kalpo arī kā līdzeklis nākotnes uzlabojumiem, identificējot biežāk sastopamās problēmas un tendences. Šī dokumentācijas sadaļa ir būtiska, lai garantētu, ka preču mazumtirdzniecības e-veikals "Piffdeals" darbojas saskaņā ar visām tehniskajām un biznesa prasībām, nodrošinot augstas kvalitātes pakalpojumu saviem lietotājiem. Skatīt testēšanas žurnālu.

Testēšanas ID	Datums	Testpiemēra ID	Testpiemēra nosaukums	Testētājs	Statuss	Kļūdas ziņojums	Kļūdas ziņojuma Nr.
Black Box							
TZ.KL.01	02.06.2024	TP.KL.01	Aktīvu kategoriju ielāde	Markuss Balodis	Veiksmīgs		
TZ.KL.02	02.06.2024	TP.KL.02	Preču skaits meklēšanas joslā	Markuss Balodis	Veiksmīgs		
TZ.KL.03	02.06.2024	TP.KL.03	Preču filtrēšana caur meklēšanas joslu	Markuss Balodis	Veiksmīgs		
TZ.KL.04	02.06.2024	TP.KL.04	Kontaktu veidlapas aizpildīšana	Markuss Balodis	Veiksmīgs		
TZ.KL.05	10.06.2024	TP.KL.04.	Kontaktu veidlapas aizpildīšana	Markuss Balodis	Neveiksmīgs	Formu nevar aizpildīt bez Google reCAPTCHA pārbaudes	KZ.01.
TZ.KL.06	10.06.2024	TP.KL.P.05	Profilu sadaļas piekļuve	Markuss Balodis	Neveiksmīgs	Ievadot adreses joslā <i>/profile</i> URL, tieku novirzīts uz <i>/login</i> lapu	KZ.02
TZ.KL.07	10.06.2024	TP.KL.P.05	Profilu sadaļas piekļuve	Markuss Balodis	Veiksmīgs		
TZ.KL.08	10.06.2024	TP.KL.P.06	Konta e-pasta vai vārda, uzvārda maiņa profila sadaļā	Markuss Balodis	Veiksmīgs		
TZ.KL.09	10.06.2024	TP.KL.P.06	Konta e-pasta vai vārda, uzvārda maiņa profila sadaļā	Markuss Balodis	Neveiksmīgs	E-pasts ir aizņemts	KZ.03
TZ.KL.10	10.06.2024	TP.KL.P.06	Konta e-pasta vai vārda, uzvārda maiņa profila sadaļā	Markuss Balodis	Neveiksmīgs	Ievades lauks pieņem tikai e-pasta tipu (back-end validācijas kļūda)	KZ.04
TZ.KL.11	10.06.2024	TP.KL.P.07	Konta paroles maiņa profila sadaļā	Markuss Balodis	Veiksmīgs		
TZ.KL.12	10.06.2024	TP.KL.P.07	Konta e-pasta vai vārda, uzvārda maiņa profila sadaļā	Markuss Balodis	Neveiksmīgs	Ievadītā parole nesakrīt ar pašreizējo	KZ.05
TZ.KL.13	10.06.2024	TP.KL.P.07	Konta e-pasta vai vārda, uzvārda maiņa profila sadaļā	Markuss Balodis	Neveiksmīgs	Jaunā parole nesakrīt ar apstiprināto	KZ.06

TZ.KL.14	10.06.2024	TP.KL.P.08	Konta dzēšana sistēmā	Markuss Balodis	Veiksmīgs		
TZ.KL.15	10.06.2024	TP.KL.P.09	Paroles atiestatīšana /login lapā	Markuss Balodis	Veiksmīgs		
TZ.KL.16	10.06.2024	TP.KL.P.09	Paroles atiestatīšana /login lapā	Markuss Balodis	Neveiksmīgs	Token atslēgas derīguma termiņš ir beidzies	KZ.07
TZ.KL.17	10.06.2024	TP.KL.P.09	Paroles atiestatīšana /login lapā	Markuss Balodis	Neveiksmīgs	E-pasts datubāzē nav atrodams	KZ.08
TZ.KL.18	10.06.2024	TP.KL.P.10	Konta e-pasta verifikācija	Markuss Balodis	Veiksmīgs		
TZ.KL.PAT.19	10.06.2024	TP.KL.PAT.11	Preču atlase no kategoriju grupas	Markuss Balodis	Neveiksmīgs	Grupā nav nevienu atribūtu	KZ.09
TZ.KL.PAT.20	10.06.2024	TP.KL.PAT.11	Preču atlase no kategoriju grupas	Markuss Balodis	Veiksmīgs		
TZ.KL.PR.21	10.06.2024	TP.KL.PR.13	Preču pievienošana grozā	Markuss Balodis	Veiksmīgs		
TZ.KL.PR.22	10.06.2024	TP.KL.PR.13	Preču pievienošana grozā	Markuss Balodis	Neveiksmīgs	Klienta kontam nav verificēta e-pasta adrese	KZ.10
TZ.AP.23	10.06.2024	TP.AP.17	Jaunu preču izveidošana	Markuss Balodis	Veiksmīgs		
TZ.AP.24	10.06.2024	TP.AP.18	Jaunu preču izveidošana	Markuss Balodis	Veiksmīgs		

Testēšanas žurnāls

7. Secinājumi

Projekta kompleksitāte un mācību pieredze:

Sākmā šķita, ka e-veikala sistēmu izstrāde būs vienkāršs uzdevums, taču projekta izstrādes gaitā sapratu, ka klientu puses funkcionalitāte ir tikai aisberga redzamā daļa. Administratora puses izstrāde prasīja dziļāku tehnisko izpratni un sarežģītāku loģiku. Šis projekts bija izaicinājums, kas man ļāva gūt būtiskas zināšanas par mūsdienīgu tīmekļa izstrādes tehnoloģiju – VILT steku, īpaši uzsverot to, cik efektīvi ar Laravel var tikt veidotas sarežģītas tīmekļa vietnes.

Tehnoloģiju izpratne:

Projekta izstrādē iegūtā pieredze ar VILT steku atklāja daudz jaunu perspektīvu, īpaši par Laravel spējām un tā saskaņošanu ar front-end tehnoloģijām kā VueJS. Izpratne par VueJS un React atšķirībām palīdzēja saprast, kāpēc dažādos projektos varētu izvēlēties vienu vai otru tehnoloģiju. VueJS, ar tā ātrumu un efektivitāti, bija īpaši piemērots šim projektam, savukārt React labākā lasāmība un pielāgojamība atstāja telpu nākotnes projektiem.

Plānošana un nākotnes izaicinājumi:

Projekts atklāja potenciālu jaunām funkcionalitātēm, piemēram, vēlmju sarakstu un datu bāzes normalizāciju, kā arī nepieciešamību mainīt lapas dizainu, izmantojot mūsdienīgus UI komponentus kā Ant Design un AG Grid. Turklāt, plānots integrēt maksājumu sistēmas kā Stripe vai Klix un savienot piegādātāju API, piemēram, Omniva un DPD.

Motivācija un turpmākā attīstība:

Projekts motivēja testēt citus izstrādes rīkus un apsvērt noSQL datu bāzu izmantošanu, kas varētu piedāvāt elastīgākas datu struktūrēšanas un mērogošanas iespējas. Projekta gaitā arī radās interese iepazīt plašāk konkurentus un veikt to darbības analīzi, lai identificētu potenciālās priekšrocības un trūkumus.

Šī pieredze ir būtiski veicinājusi manas prasmes un zināšanas kā izstrādātājam, atklājot jaunas tehnoloģijas un metodoloģijas, kuras noderēs nākotnes projektos. Tā ir bijusi lieliska iespēja augt profesionāli un sagatavot pamatu turpmākai karjerai IT nozarē.

8. Lietoto terminu un saīsinājumu skaidrojumi

Tehniskā dokumentācijā tiek izmantoti termini un saīsinājumi, kas veicina dokumenta saprotamību. Šī sadaļa ir izveidota kā informatīvs resurss visiem dokumentācijas lasītājiem, neatkarīgi no to saistības ar IT nozari. Sadaļā tiek apkopoti visi dokumentā lietotie termini jeb jēdzieni un saīsinājumi. Tie ir sakārtoti tabulas veidā, alfabētiskā secībā, nodrošinot skaidru un strukturētu pieeju. Dokumentā izmantotie termini un saīsinājumi tiek izcelti ar slīprakstas vai iekavu formātu, kas atvieglo to atpazīšanu un izpratni. Skatīt 1. un 2. tabulas, lai gūtu priekšstatu par tās definīcijas nozīmi.

1. tabula

Saīsinājumu tabula

Saīsinājums	Apraksts
API	Komandu un protokolu kopums, kas paredzēts citu programmatūras mijiedarbībai
CRUD	Četras pamatfunkcijas datu pārvaldībai datubāzēs – izveidot, lasīt, rediģēt un dzēst
CSRF	Mājaslapas uzbrukumu veids, kurā uzbrucējs sūta viltus pieprasījumus no uzticama, autorizēta lietotāja
CSS	Stilu lapas valoda, kas izveido mājaslapas vizuālo noformējumu
HTML	Hiperteksta marķējuma valoda, ko izmanto mājaslapu struktūras izveidei
HTTP	Interneta protokols, kas izmanto datus pārraidīšanai tīmekļa pārlūkprogrammās
IDE	Programmatūra, kas nodrošina visaptverošus rīkus programmatūras izstrādei
JSON	Teksta formāts datu apmaiņai, kas balstīts uz JavaScript programmēšanas valodas
MVC	Programmēšanas arhitektūras modelis, ko izmanto Laravel, React, Angular u.c.
RGB	Krāsu modelis, ko izmanto attēlu un ekrāna krāsu noteikšanai
HEX	HEX kodi ir heksadecimāls formāts krāsu identificēšanai.
SEO	Tehnikas rīki un stratēģijas, lai uzlabotu vietnes redzamību meklētājprogrammās
SKU	Krājuma uzskaites vienība, unikāls identifikators precei noliktavas pārvaldībā
SPA	Tīmekļa lietotnes, kas ielādējas vienā lapā, nodrošinot gludu lietotāju pieredzi
SQL	Strukturētu pieprasījumu valoda datubāzu pārvaldībai
UML	Grafiska valoda sistēmu projektēšanai un analizēi
XSS	Mājaslapu uzbrukumu veids, kas ļauj ievietot kaitīgu kodu tīmekļa lapās
KP	Klienta puse
TZ	Testēšanas žurnāls
TP	Testpiemērs
PAT	Preču atlase
P	Profils
AP	Administrācijas panelis
PR	Prece

2. tabula

Lietoto terminu tabula

Termins	Apraksts
Back-end	Servera puses daļa, kas apstrādā datu bāzi un lietotāja pieprasījumus
Bcrypt	Algoritms drošu parolu šifrēšanai, plaši izmantots modernās mājaslapās
Black box	Programmatūras testēšanas metode
Debugging	Atklādošanas process, kurā tiek meklētas un labotas programmatūras kļūdas
Eloquent ORM	Izmantots Laravel ietvarā, kas atvieglo datubāzes operāciju veikšanu, izmantojot objektu sintaksi
ES moduļi	JavaScript moduļi, kas ļauj veikt kodu sapludināšanu un atkārtotu izmantošanu

Favicon	Mazs attēls pārlūkprogrammas cilnes augšpusē, kas tiek izmantots kā mājaslapu identifikators
Framework	Ietvars. Programmēšanas struktūra vai platforma, kas nodrošina gatavu kodu bāzi un rīkus aplikāciju izstrādei
Front-end	Klienta puses daļa, kas attiecas uz lietotāja saskarni un vizuālo noformējumu
GitHub	Serviss, kas nodrošina kodu glabāšanu un pārvaldību, izmantojot Git versiju kontroles sistēmu
Web host	Pakalpojuma sniedzējs, kas nodrošina mājaslapu serveru resursa izvietojumam
Web hosting platform	Platforma vai pakalpojums, kas sniedz mājaslapu uzturēšanu uz servera
Hover	Peles kursora atrašanās vieta virs kāda tīmekļa objekta vai elementa bez klikšķa
UI komponentes	Atkārtoti izmantojami vizuālie elementi, piemēram, pogas un izvēlnes, kas veido lietotāja saskarni
localStorage	Pārlūkprogrammas sniegta iespēja saglabāt lietotāja datus lokāli klienta ierīcē bez termiņa ierobežojuma, kas ļauj datiem palikt pārlūka sesijā starpā
Modal	Uznirstošais logs, kas parādās pār tīmekļa lapas saturu, lai piedāvātu informāciju vai prasītu lietotāja iejaukšanos
onClick	Notikums programmēšanā, kas tiek aktivizēts, kad lietotājs noklikšķina uz kāda elementa, piemēram, pogas vai saites
Plugins	Spraudņi. Papildinājumi vai moduļi, kas pievienoti programmatūrai, lai paplašinātu tās funkcionalitāti
Routes	Definīcijas programmēšanas ietvaros, kas norāda, kāda veida pieprasījumiem (piemēram, URL) ir jāatbild un kāda kontroliera funkcija tiek izsaukta atbildei
Stripe	Maksājumu apstrādes platforma, kas tiek integrēta tīmekļa vietnēs un aplikācijās, lai ļautu veikt drošus tiešsaistes maksājumus
White box	Programmatūras testēšanas metode

9. Literatūras un informācijas avotu saraksts

Tehniskās dokumentācijas un sistēmas izstrādē tiek pielietoti dažādi informācijas resursi. Šī nodaļa ir izveidota, lai apkopotu visus izmantotos resursus, kuri pilnveido sistēmu un tās tehnisko dokumentāciju. Norādītās hipersaites ir tikai daļa no visiem izmantotiem resursiem izstrādes laikā. Skatīt 3. tabulu

3. tabula

Informācijas avotu tabula

Nosaukums	Hipersaite	Lietošanas datums
Bagisto Demo	https://demo.bagisto.com/bagisto-common/	2023-09-10
Bagisto mājaslapa	https://bagisto.com/en/	2023-10-05
Composer mājaslapa	https://getcomposer.org/	2024-01-15
Draw.io diagrammas	https://app.diagrams.net/	2024-04-01
Fixer.io	https://fixer.io/	2024-03-22
GitHub repozitorija Laravel tulkošanas pakotnei	https://github.com/tohidplus/laravel-vue-translation	2023-09-20
Inertia.js dokumentācija	https://inertiajs.com/	2023-11-22
Laravel dokumentācija	https://laravel.com/docs/11.x/	2023-10-28
Laravel Daily - VueJS tulkošanas spraudnis	https://laraveldaily.com/post/laravel-and-vuejs-translations-with-i18n-plugin	2024-03-30
Laravel diskusija par pagināciju	https://laracasts.com/discuss/channels/laravel/jeffrey-way-how-did-you-do-the-pagination-in-laracast	2023-12-06
Laravel diskusija par meklēšanas joslu	https://laracasts.com/discuss/channels/general-discussion/how-does-a-search-bar-works	2024-02-11
Laravel diskusija par Vue pret React	https://laracasts.com/discuss/channels/laravel/vue-js-or-react-js-for-laravel	2023-11-30
Laravel diskusija par Ziggy drošību	https://laracasts.com/discuss/channels/vue/is-ziggy-a-security-problem-how-should-i-prevent-it-to-list-all-routes	2023-12-14
MVC arhitektūras izpratne	https://fkrihnif.medium.com/understanding-the-mvc-architecture-in-laravel-a-comprehensive-guide-8f620cc139b6	2024-01-30
SQL Datubāze definīcija	https://www.solarwinds.com/resources/it-glossary/sql-database	2023-09-15
Stack Overflow Kontrolieru un modeļu ģenerēšana	https://stackoverflow.com/questions/14265996/generate-controller-and-model	2024-02-08
Tailwind CSS Dokumentācija	https://v2.tailwindcss.com/docs	2024-03-25
Vite	https://vitejs.dev/	2023-10-18

Pielikums

```

16 class GoodsController extends Controller
17 {
18     private function getActiveData() { // Funkcija, lai iegūtu aktīvos datus kategorijām, grupām un atribūtiem
19         return [
20             'activeCategories' => Category::where('status', 'Aktīvs')->get(),
21             'activeGroups' => Group::where('status', 'Aktīvs')->get(),
22             'activeAttributes' => Attribute::where('status', 'Aktīvs')->get(),
23         ];
24     }
25
26     public function index(Request $request) // Parāda preču sarakstu
27     {
28         $search = $request->input('search', ''); // Iegūst meklēšanas ievadi
29         $status = $request->input('status'); // Iegūst statusa ievadi
30
31         $query = Good::with(['group.category', 'attribute']); // Veido jautājumu ar grupas un atribūta attiecībām
32
33         if (!empty($search)) { // Ja meklēšanas lauks nav tukšs
34             $query->where(function ($q) use ($search) {
35                 $q->where('name', 'like', "%{$search}%") // Meklē pēc nosaukuma
36                 ->orWhere('sku', 'like', "%{$search}%") // Vai pēc SKU
37                 ->orWhereHas('attribute', function ($subQuery) use ($search) {
38                     $subQuery->where('name', 'like', "%{$search}%"); // Vai pēc atribūta nosaukuma
39                 });
40             });
41         }
42
43         if (!empty($status)) { // Ja statusa lauks nav tukšs
44             $query->where('status', $status); // Pievieno statusa filtru
45         }
46
47         $goods = $query->orderBy('created_at', 'desc')->paginate(15); // Sakārto pēc izveides datuma un sadala lapās
48         $activeData = $this->getActiveData(); // Iegūst aktīvos datus
49         return Inertia::render('Admin/Goods', array_merge([
50             'goods' => $goods,
51             'filters' => $request->only('search', 'status'),
52             'totalGoods' => $goods->total(),
53         ], $activeData)); // Atgriež skatu ar datiem
54     }
55
56     public function store(Request $request) // Veido jaunu preci
57     {
58         $validated = $request->validate([
59             'sku' => 'required|string|max:255|unique:goods,sku', // Pārbauda, vai SKU ir unikāls
60             'status' => 'required|in:Aktīvs,Deaktivizēts', // Pārbauda, vai statuss ir derīgs
61             'group_id' => 'nullable|exists:groups,id', // Pārbauda, vai grupas ID eksistē
62             'attribute_id' => 'nullable|exists:attributes,id', // Pārbauda, vai atribūta ID eksistē
63         ]);
64
65         $validated['name'] = $request->input('name', 'Nenosaukta prece'); // Ja nosaukums nav norādīts, piešķir "Nenosaukta prece"
66         $validated['description'] = $request->input('description', ''); // Apraksts
67         $validated['image'] = $request->input('image', ''); // Attēls
68         $validated['stock_quantity'] = $request->input('stock_quantity', 0); // Noliktavas daudzums
69         $validated['price'] = $request->input('price', 0.00); // Cena
70
71         $good = Good::create($validated); // Izveido preci ar validētajiem datiem
72
73         return redirect()->route('admin.goods.edit', $good->id); // Pārdresē uz preces rediģēšanas lapu
74     }
75
76     public function edit($id) // Parāda preces rediģēšanas skatu
77     {
78         $good = Good::with(['group.category', 'attribute'])->findOrFail($id); // Atrod preci ar attiecībām
79         $activeData = $this->getActiveData(); // Iegūst aktīvos datus
80
81         return Inertia::render('Admin/GoodDetail', array_merge([
82             'good' => $good,
83         ], $activeData)); // Atgriež skatu ar datiem
84     }
85 }

```

Preču CRUD Eloquent ORM back-end kods

```

9
10 class CartController extends Controller
11 {
12     public function add(Request $request) // Pievieno preci grozam
13     {
14         $user = Auth::user(); // Iegūst pašreizējo lietotāju
15         if (!$user->hasVerifiedEmail()) { // Pārbauda, vai lietotājam ir apstiprināts e-pasts
16             return redirect()->route('verification.notice'); // Ja nē, pārdresē uz verifikācijas paziņojumu
17         }
18         $cart = $user->cart()->firstOrCreate(['user_id' => $user->id]); // Iegūst vai izveido grozu lietotājam
19         $existingItem = $cart->items()->where('good_id', $request->input('goodId'))->first(); // Pārbauda, vai prece jau eksistē grozā
20         if ($existingItem) { // Ja prece jau eksistē grozā
21             if ((int) $request->input('quantity') !== $existingItem->quantity) { // Ja daudzums ir mainījies
22                 $existingItem->update(['quantity' => $request->input('quantity')]); // Atjaunina daudzumu
23                 Log::info('Quantity updated for item ID: ' . $existingItem->id); // Pieraksta logā
24             }
25         } else {
26             $cart->items()->create([ // Ja prece nav grozā, pievieno to
27                 'good_id' => $request->input('goodId'),
28                 'quantity' => $request->input('quantity')
29             ]);
30         }
31         return back()->with('success', 'Cart updated successfully'); // Atgriež atbildi ar veiksmīgu ziņojumu
32     }
33     public function getCartItems() // Iegūst groza preces
34     {
35         $user = Auth::user(); // Iegūst pašreizējo lietotāju
36         $cartItems = $user->cart->items()->with('good')->get() ?? []; // Iegūst preces no groza ar saistītajiem labumiem
37         return response()->json($cartItems); // Atgriež preces kā JSON atbildi
38     }
39     public function index() // Parāda groza skatu
40     {
41         $user = Auth::user(); // Iegūst pašreizējo lietotāju
42         $cartItems = $user->cart->items()->with('good')->get() ?? []; // Iegūst preces no groza ar saistītajiem labumiem
43         return Inertia::render('Cart', [
44             'cartItems' => $cartItems,
45         ]); // Atgriež Inertia skatu ar groza precēm
46     }
47     public function remove($id) // Noņem preci no groza
48     {
49         $user = Auth::user(); // Iegūst pašreizējo lietotāju
50         $cartItem = $user->cart->items()->find($id); // Atrod preci grozā pēc ID
51         if ($cartItem) { // Ja prece ir atrasta
52             $cartItem->delete(); // Dzēš preci no groza
53         }
54         return $this->index(); // Atgriež atjauninātu groza skatu
55     }
56     public function updateAndCheckout(Request $request) // Atjaunina grozu un pāriet uz norēķinu
57     {
58         $user = Auth::user(); // Iegūst pašreizējo lietotāju
59         $cart = $user->cart; // Iegūst lietotāja grozu
60         foreach ($request->input('items') as $item) { // Iterē caur precēm no pieprasījuma
61             $cartItem = $cart->items()->find($item['id']); // Atrod preci grozā
62             if ($cartItem) { // Ja prece ir atrasta
63                 $cartItem->update(['quantity' => $item['quantity']]); // Atjaunina daudzumu
64             }
65         }
66         return redirect()->route('checkout.index'); // Pārdresē uz norēķinu lapu
67     }
68 }
69

```

Groza Eloquent ORM kods


```

94 // Navigation data routes
95 Route::prefix('navigation-data')->group(function () {
96     Route::get('/categories', [NavigationDataController::class, 'getActiveCategories'])->name('navigation.data.categories');
97     Route::get('/goods-count', [NavigationDataController::class, 'getActiveGoodsCount'])->name('navigation.data.goodsCount');
98     Route::get('/active-goods', [NavigationDataController::class, 'getActiveGoods'])->name('navigation.data.activeGoods');
99     Route::get('/search-goods', [NavigationDataController::class, 'searchGoods'])->name('navigation.data.searchGoods');
100 });
101
102
103 Route::get('/goods/{id}', [GoodsPreviewController::class, 'show'])->name('goods.show');
104 Route::get('/goods', [FilterGoodsController::class, 'index'])->name('goods.filter');
105
106 Route::prefix('admin-goods')->middleware(['auth', 'checkAdminRole'])->group(function () {
107     Route::get('/', [GoodsController::class, 'index'])->name('admin.goods.index');
108     Route::post('/store', [GoodsController::class, 'store'])->name('admin.goods.store');
109     Route::get('/{id}/edit', [GoodsController::class, 'edit'])->name('admin.goods.edit');
110     Route::patch('/{id}', [GoodsController::class, 'update'])->name('admin.goods.update');
111     Route::get('/{id}', [GoodsController::class, 'show'])->name('admin.goods.show');
112     Route::post('/upload-image', [GoodsController::class, 'uploadImage'])->name('admin.goods.upload-image');
113     Route::delete('/{id}', [GoodsController::class, 'destroy'])->name('admin.goods.destroy');
114 });
115
116 Route::prefix('admin-attributes')->middleware(['auth', 'checkAdminRole'])->group(function () {
117     Route::get('/', [AttributesGoodsController::class, 'index'])->name('admin.attributes.index');
118     Route::post('/store', [AttributesGoodsController::class, 'store'])->name('admin.attributes.store');
119     Route::patch('/admin-attributes/{id}', [AttributesGoodsController::class, 'update'])->name('admin.attributes.update');
120     Route::delete('/{id}', [AttributesGoodsController::class, 'destroy'])->name('admin.attributes.destroy');
121 });
122
123 Route::prefix('admin-groups')->middleware(['auth', 'checkAdminRole'])->group(function () {
124     Route::get('/', [GroupGoodsController::class, 'index'])->name('admin.groups.index');
125     Route::post('/store', [GroupGoodsController::class, 'store'])->name('admin.groups.store');
126     Route::patch('/admin-groups/{id}', [GroupGoodsController::class, 'update'])->name('admin.groups.update');
127     Route::delete('/{id}', [GroupGoodsController::class, 'destroy'])->name('admin.groups.destroy');
128 });
129
130 Route::prefix('admin-categories')->middleware(['auth', 'checkAdminRole'])->group(function () {
131     Route::get('/', [CategoryGoodsController::class, 'index'])->name('admin.categories.list');
132     Route::post('/store', [CategoryGoodsController::class, 'store'])->name('admin.goods.storeCategory');
133     Route::patch('/admin-categories/{id}', [CategoryGoodsController::class, 'update'])->name('admin.goods.updateCategory');
134     Route::delete('/admin-categories/{id}', [CategoryGoodsController::class, 'destroy'])->name('admin.goods.destroyCategory');
135 });
136
137 Route::prefix('admin-users')->name('admin-users')->group(function () {
138     Route::get('/', [ClientAccountController::class, 'index'])->name('index')->middleware('checkAdminRole');
139     Route::post('/store', [ClientAccountController::class, 'storeAdmin'])->name('storeAdmin')->middleware('checkAdminRole');
140     Route::get('/{id}', [ClientAccountController::class, 'show'])->name('show')->middleware('checkAdminRole');
141     Route::post('/password/email', [ClientAccountController::class, 'sendResetLinkEmail'])->name('password.email')->middleware('checkAdminRole');
142     Route::patch('/update-status', [ClientAccountController::class, 'updateStatus'])->name('updateStatus')->middleware('checkAdminRole');
143     Route::delete('/delete', [ClientAccountController::class, 'deleteUsers'])->name('delete')->middleware('checkAdminRole');
144 });
145
146 // Auth routes
147 require __DIR__ . '/auth.php';

```

InertiaJS saites (routes) kods, front-end savienojums ar back-end

Izstrādātā sistēma pieejama versionēšanas sistēmā GitHub:

<https://github.com/vFar/piffdeals>