

Liepājas Valsts tehnikums

**Preču mazumtirdzniecības e-veikals “Piffdeals”**

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija **33484011 Programmēšana**

Grupas nosaukums **Programmēšanas tehniķis**

Projekta izstrādātājs

/vārds, uzvārds, paraksts/

Eksāmena datums 202.\_\_. gada \_\_\_\_\_\_\_\_\_\_\_\_\_

Liepāja 202\_

**Saturs**

[Ievads 4](#_Toc168259998)

[1.Uzdevuma formulējums 5](#_Toc168259999)

[2. Programmatūras prasību specifikācija 6](#_Toc168260000)

[2.1. Produkta perspektīva 6](#_Toc168260001)

[2.2. Sistēmas funkcionālās prasības 6](#_Toc168260002)

[2.2.1. Klienta sadaļas lapa 6](#_Toc168260003)

[2.2.2. Pieejamās preces 11](#_Toc168260004)

[2.2.3. Lietotāja konts 12](#_Toc168260005)

[2.2.4. Administrācijas panelis 16](#_Toc168260006)

[2.2.5. Citi 19](#_Toc168260007)

[2.3. Sistēmas nefunkcionālās prasības 21](#_Toc168260008)

[2.4. Gala lietotāja raksturiezīmes 21](#_Toc168260009)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 22](#_Toc168260010)

[1.1. Izvēlēto risinājuma līdzekļu un valodu apraksts 22](#_Toc168260011)

[1.1.1. Back-end puses programmēšanas valodas līdzekļi 22](#_Toc168260012)

[3.1.2. Front-end puses programmēšanas valodas līdzekļi 23](#_Toc168260013)

[1.1.2. Izmantotie rīki 24](#_Toc168260014)

[3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts 25](#_Toc168260015)

[3.3. Līdzekļu un rīku lietojuma pamatojums 26](#_Toc168260016)

[4. Sistēmas modelēšana un projektēšana 27](#_Toc168260017)

[4.1. Sistēmas struktūras modelis 27](#_Toc168260018)

[4.1.1. ER diagramma 27](#_Toc168260019)

[4.2. Funkcionālais un dinamiskais sistēmas modelis 29](#_Toc168260020)

[4.2.1. Aktivitāšu diagramma (Activity) 29](#_Toc168260021)

[4.2.2. Lietojumgadījumu diagramma (Use Case) 31](#_Toc168260022)

[4.3. Sistēmas moduļu apraksts un algoritmu shēmas 33](#_Toc168260023)

[4.3.1. Google OAuth pieslēgšanās algoritmu shēma 33](#_Toc168260024)

[4.3.2. Paroles atjaunošanas algoritms 35](#_Toc168260025)

[5. Lietotāju ceļvedis 36](#_Toc168260026)

[5.1. Reģistrācija e-veikalā kā viesis 36](#_Toc168260027)

[5.2. Sazināšanās ar administratoriem 37](#_Toc168260028)

[6. Testēšanas dokumentācija 39](#_Toc168260029)

[6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums 39](#_Toc168260030)

[6.2. Alternatīvās testēšanas metodes un rīki 39](#_Toc168260031)

[6.3. Testpiemēru kopa 40](#_Toc168260032)

[6.4. Testēšanas žurnāls 40](#_Toc168260033)

[7. Secinājumi 41](#_Toc168260034)

[8. Lietoto terminu un saīsinājumu skaidrojumi 42](#_Toc168260035)

[9. Literatūras un informācijas avotu saraksts 44](#_Toc168260036)

[Pielikums 45](#_Toc168260037)

# 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, kā arī dokumenta pielikuma sadaļā būs uzskaitīti visi intervijas jautājumi un intervijas protokoli.

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 autentifikācijas 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 jāsazinās 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 preci izsūtīt caur citiem piedāvātiem piegādātājiem, piemēram, Omniva, DPD u.t.t.

Preču mazumtirdzniecība ir guvusi krietnu popularitāti kopš 2020. gada COVID-19 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 rietumpreces.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 atttiecināmā poga

Apstrāde:

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

Izvaddati:

E-veikala klienta lietotājam tiek izvadīts sekmīgs paziņojums

**P.2. Preču kategoriju ielāde sākumlapā**

Mērķis:

Preču kategoriju ielādi sākumlapā, galvenes sadaļā

Ievaddati:

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

Apstrāde:

Sākumlapas 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 sākumlapā**

Mērķis:

Uzrādīt lapas klientam kategoriju grupas

Ievaddati:

E-veikala lietotājam ir jāapmeklē sākumlapa 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, sākumlapā, 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 sākumlapā**

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 jāsastāv 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 attieciešie 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 uzskaite meklēšanas joslā**

Mērķis:

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

Ievaddati:

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

Apstrāde:

Funckija atlasa visas preces datubāzē, kam statuss ir “Aktīvs”

Izvaddati:

Lapu pārlādēšanas gadījumā, galvenes meklešanas 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 ievadīšanas, 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 galvenas 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 statistikas 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.

**P.13. Reklāmu baneri**

Mērķis:

Nodrošināt reklāmu baneru karuseli ar 5 reklāmām, kuras administrators var pārvaldīt *(CRUD)* un attēli tiek ielādēti no datubāzes.

Ievaddati:

Administratoram jāizveido un jāpārvalda reklāmas baneri datubāzē.

Apstrāde:

Funkcija ielādē reklāmas banerus no datubāzes un attēlo tos karuselī, kas spēj veikt CRUD operācijas.

Izvaddati:

Karuselis ar 5 reklāmām tiek attēlots klientiem lapā.

**P.14. Jaunākās preces**

Mērķis:

Attēlot sadaļu ar jaunākajām pievienotajām precēm, kurām ir statuss "Aktīvs".

Ievaddati:

Administratoram jāatzīmē jaunās preces ar statusu "Aktīvs".

Apstrāde:

Funkcija atlasa un ielādē preces ar statusu "Aktīvs" no datubāzes.

Izvaddati:

Sadaļa ar jaunākajām aktīvajām precēm tiek attēlota klientiem lapā.

### 2.2.2. Pieejamās preces

**P.15. 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

### 2.2.3. Lietotāja konts

**P.16. Konta pieslēģšanās**

Mērķis:

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

Ievaddati:

Lietotājam ir jāreģistrē sev konts. 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.17. 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.18. 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.19. 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.20. Paroles atjaunošana pēc vēstules saņemšanas**

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 lauks 1 automātiski tiek aizpildīts ar viņa 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.

Izvaddati:

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

**P.21. Konta reģistrācija, lietojot Facebook OAuth API**

Mērķis:

Nodrošināt lietotājiem iespēju reģistrēt savu kontu, izmantojot Facebook API.

Ievaddati:

Lietotājam jāapmeklē reģistrācijas lapa un jāuzspiež uz pogas “Reģistrēties ar Facebook”.

Apstrāde:

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

Izvaddati:

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

**P.22. 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 sākumlapu ar piešķirtu sesijas kodu.

**P.23. 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.24. Konta informācijas atjaunināšana**

Mērķis:

Sniegt lietotājam iespēju atjaunināt savu konta informāciju.

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 jaunos datus savā profilā.

**P.25. 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 dzēš pašreizējo sesiju un anonimizē lietotāja kontu datubāzē

Izvaddati:

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

**P.26. Lietotāja pasūtījumu vēstures apskate**

Mērķis:

Nodrošināt lietotājam iespēju apskatīt savu pasūtījumu vēsturi.

Ievaddati:

Lietotājam jābūt autorizētam un jāapmeklē sadaļa “Pasūtījumu vēsture”.

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.27. Lietotāja adreses pārvaldība**

Mērķis:

Nodrošināt lietotājam iespēju pievienot, rediģēt un dzēst adreses savā kontā.

Ievaddati:

Lietotājam jābūt autorizētam un jāapmeklē sadaļa “Adreses”. Jāaizpilda adrešu forma un jāapstiprina izmaiņas.

Apstrāde:

Funkcija pārbauda, vai ievadītie adrešu dati ir aizpildīti un derīgi. Ja dati ir derīgi, funkcija saglabā vai atjaunina adresi datubāzē.

Izvaddati:

Lietotājs redz savu adrešu sarakstu, un saņem paziņojumu par veiksmīgu izmaiņu veikšanu.

### 2.2.4. Administrācijas panelis

**P.28. 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.29. Administratora mega meklēšanas josla**

Mērķis:

Nodrošināt administratoriem iespēju izmantot mega meklēšanas joslu, lai ātri atrastu dažādus sistēmas elementus.

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, kategoriju nosaukumi, grupu nosaukumi, atribūtu nosaukumi, lietotāju e-pasti, lietotāju tālruņa numuri, citu administratoru kontu e-pasti u.c.

Izvaddati:

Tiek parādīti meklēšanas rezultāti, kas atbilst ievadītajiem atslēgas vārdiem.

**P.30. 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.31. 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.32. Kataloga komponentes**

**Mērķis:**

Nodrošināt administratoriem iespēju pārvaldīt preču katalogu ar datu tabulu (datatable) palīdzību, ļaujot pievienot, dzēst, rediģēt un skatīt preces 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.33. Klientu sadaļa**

**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.

**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.

### 2.2.5. Citi

**P.34. Kontaktu veidlapas aizpildīšana un nosūtīšana uz e-pastu**

Mērķis:

Sniegt iespēju e-veikala klientiem sekmīgi nosūtīt saziņas veidlapu uz administratora e-pastu: support@piffdeals.lv

Ievaddati:

Lietotājam ir jāapmeklē sadaļa “Sazinies ar mums” jeb “Kontakti”, kurā aizpilda visus nepieciešamos laukus un nospiež pogu “Sūtīt ziņu”

Apstrāde:

Funkcija pārbauda vai visi lauki ir aizpildīti un pēc pogas nospiešas aizsūta ziņu uz e-veikala atbalsta e-pastu

Izvaddati:

Tiek izvadīts sekmīgs paziņojums par veiksmīgu ziņas nosūtīšanu e-veikala administratoriem

**P.35. E-veikala tulkošana caur *JSON* rīkiem**

Mērķis:

Sniegt iespēju veikala lietotājiem tulkot tīmekļa vietni no piedāvātājām valodām. Noklusējuma valoda ir latviešu valoda

Ievaddati:

Lietotājam ir jāorientējas ar e-veikala galvas daļu un jālieto *HTML Select taga* funkcijas, lai tulkotu

Apstrāde:

Funkcija pārbauda, kura valoda tiek izvēlēta un ar *JSON* rīku palīdzību, pārlādē lapu, kurā tekstu elementi ir tulkoti

Izvaddati:

Lapa tiek tulkota pēc lietotāja izvēlētās valodas

**P.36. Gaišā vai tumšā krāsu režīma mainīšanas pēc lietotāju ieskatiem**

Mērķis:

Sniegt lietotājam iespēju mainīt monohroma krāsu pēc pašu ieskata

Ievaddati:

Lietotājam ir jāorientējas ar e-veikala galvas daļu un jālieto *HTML Button*, kurā iestata sev piemēroto krāsas režīmu

Apstrāde:

Funkcija pārbauda, kurš variants ir izvēlēts un atkarīgi no tā dinamiski pārmaina krāsu konkrētiem elementiem no gaišu uz tumšu vai otrādi

Izvaddati:

Lietotājs ir sekmīgi pārmainījis krāsu režīmu pa visu e-veikalu

**P.37. 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

## 2.3. Sistēmas nefunkcionālās prasības

1. Lietotāju un vietnes saskarne pēc noklusējuma ir latviešu valodā, toties ir iespēja tulkot e-veikalu uz angļu vai krievu valodu;
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 *RGB* 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 jāsazinās ar pakalpojuma sniedzēju gadījumā, ja vietnē ir servera problēmas;
6. E-veikals ir pielāgots uz dažāda izmēra ekrāniem;
7. 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.

Pirmās mājaslapas versijas ietver sniegt iespēju apskatīt preces un veidot pasūtījumus, neveicot apmaksi. 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 iespējamos 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.

## 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 izvēlējies šādus līdzekļus un rīkus, un to iemesli ir argumentēti:

### 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](https://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](https://mysql.com)

* **XAMPP**

XAMPP ir brīva, atvērtā 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 izvietošanas produkcijā uz oficiālu serveri. XAMPP piedāvā veikt darbības ne tikai ar MySQL, bet arī ar MariaDB. Sīkaka informācija: [apachefriends.org](https://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. Sīkāka informācija: [github.com/tighten/ziggy](https://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. Sīkāka informācija: [nodejs.org](https://nodejs.org)

### 3.1.2. Front-end puses programmēšanas valodas līdzekļi

* **VueJS**

VueJS jeb Vue.js vai Vue ir atvērtā 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 komponenšu 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 populārākājiem JavaScript *framework*, kura cieši konkurē ar React un Angular. Sīkāka informācija: [vuejs.org](https://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](https://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](https://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](https://vitejs.dev)

### 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](https://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 pielietos 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](https://phpmyadmin.net)

* **Laravel Telescope**

Elegants atkļūdošanas *(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 atkļūdošanu 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](https://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](https://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](https://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](https://php.net)

## 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.

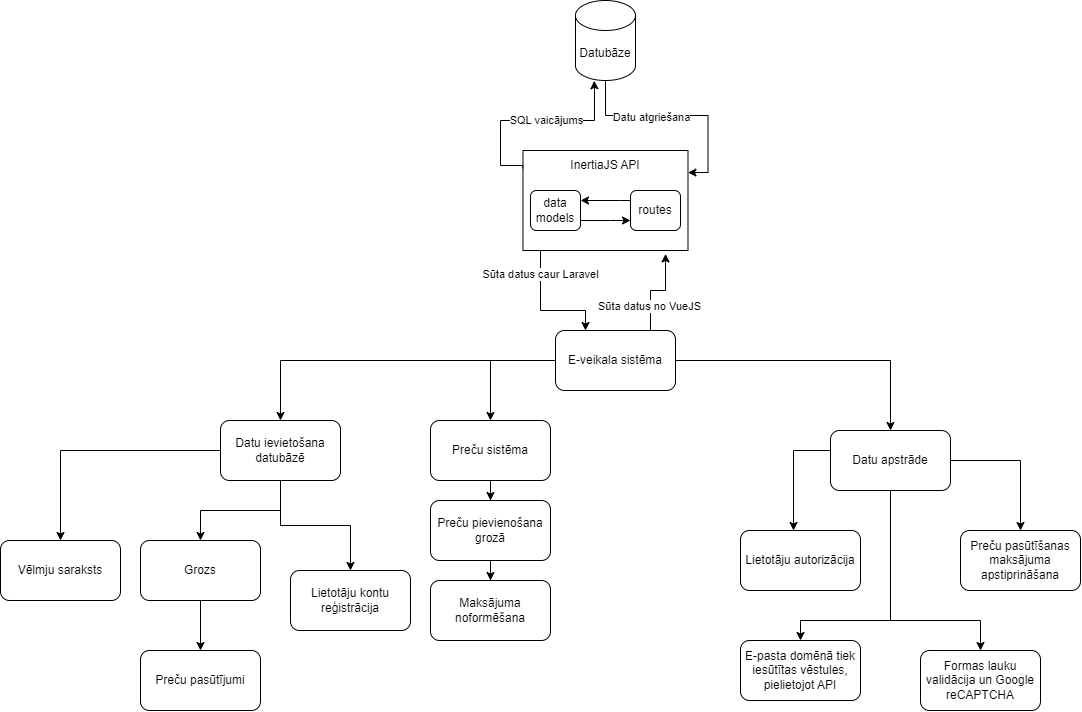
Laravel un Vue labāk sadarbojas nekā Laravel un React, piemēram, Laravel piedāvā plašāk sarakstītu dokumentāciju priekš Vue, nevis React. Tika apdomāta stratēģija, kā panākt front-end un back-end komunikāciju, tai vietā nāk InertiaJS, kas palīdz komunicēt starp abām pusēm un padot tālāk precīzu informāciju.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

Lielākā daļu līdzekļu pamatojums lietošanā saistās ar skolas un prakes 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.

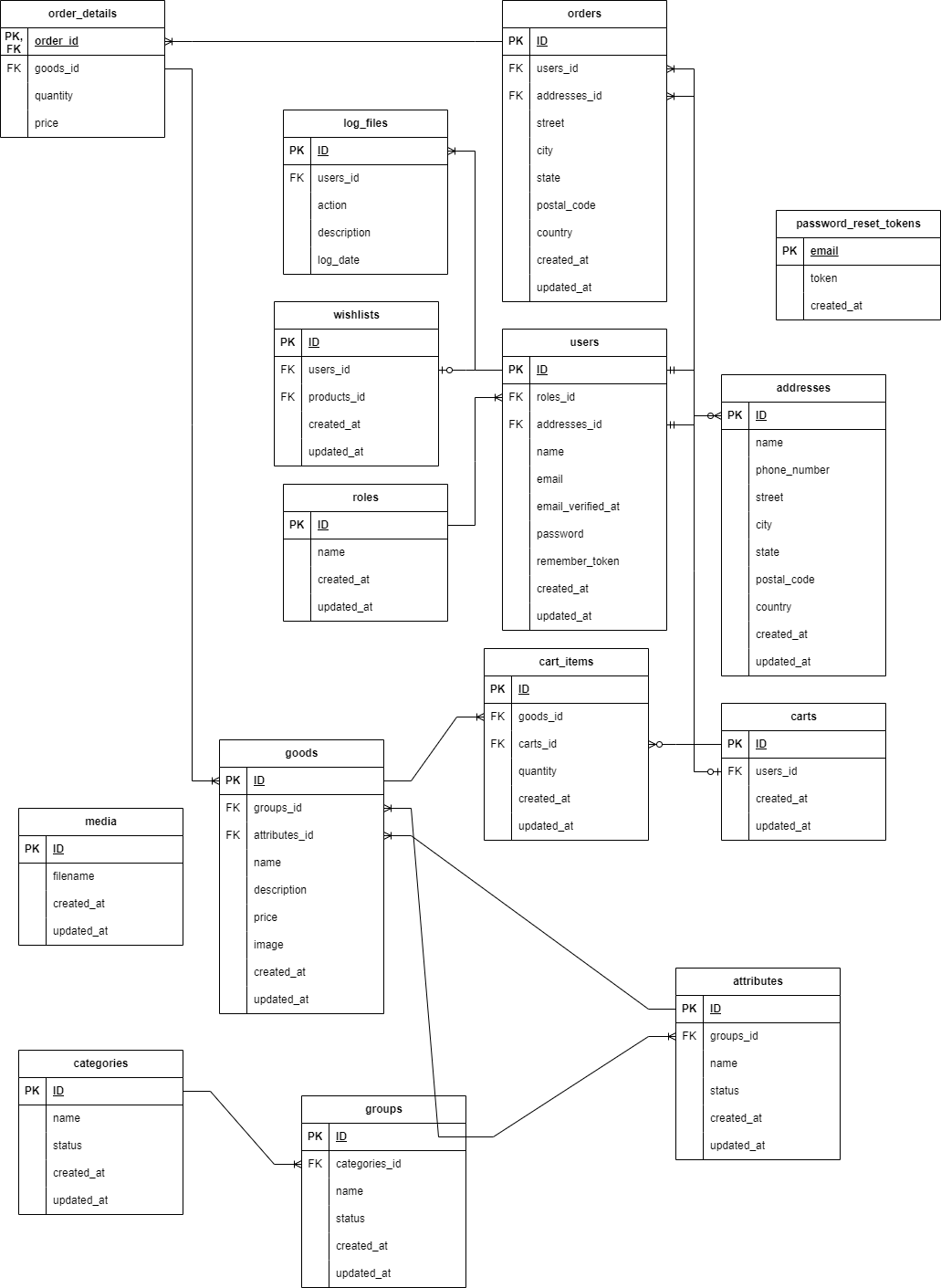
## 4.1. Sistēmas struktūras modelis

Sistēmas struktūras modelis ir attiecināms pret sistēmas komponenšu 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 sistemātiskais modelis.**

### 4.1.1. ER diagramma

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

Veidotā ER diagramma sastāv no trīspadsmit tabulām, kurā no trim ir pielietotas daudzi-pret-daudziem saišu nolūkiem, t.i. veidot organizētu saikni, veicināt datu normalizāciju un pievienot papildu atribūtus, piemēram, tabulā “goods” jeb preces. Preces var būt vairākās preču kategorijās, kamēr preču kategorijās var būt vairākas preces (skatīt 2. attēlu).

**2. attēls. Datubāzes ER diagramma**

## 4.2. Funkcionālais un dinamiskais sistēmas modelis

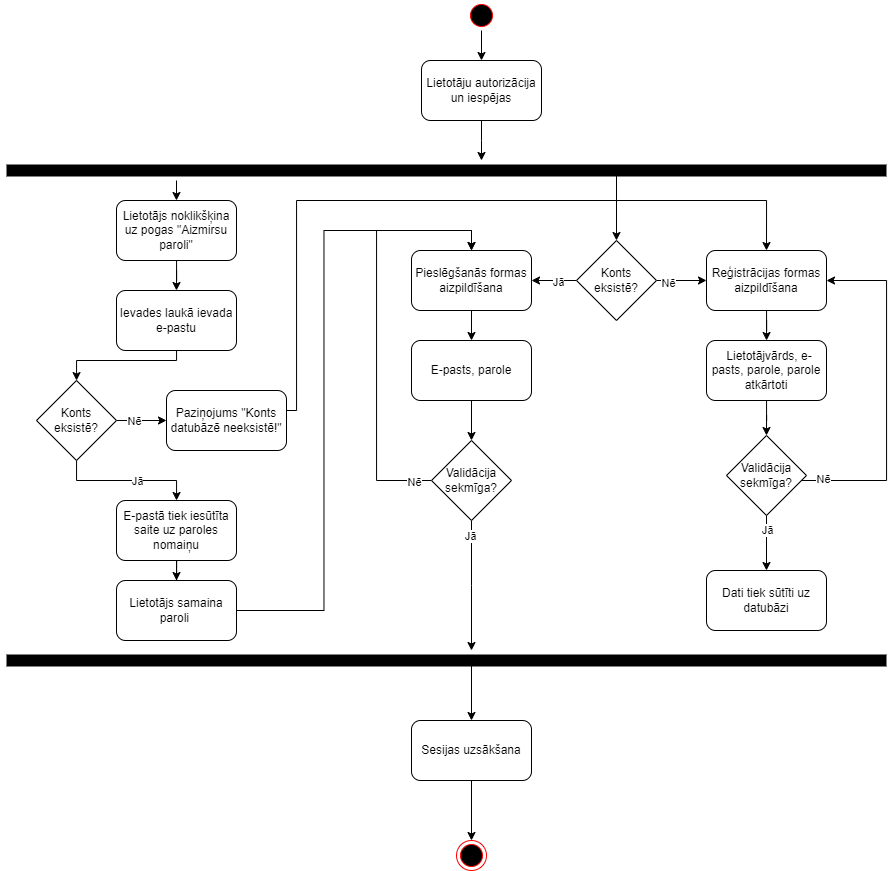
Funkcionālais un dinamiskais sistēmas modelis ir attiecināms uz sistēmas darbību un tās komponentu mijiedarbību. Šo modeļu mērķis ir palīdzēt izstrādātājiem un sistēmas analītiķiem izprast, kā sistēma darbojas un kādas ir tās komponentu attiecības. Funkcionālais modelis fokusējas uz sistēmas sniegtajām 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)

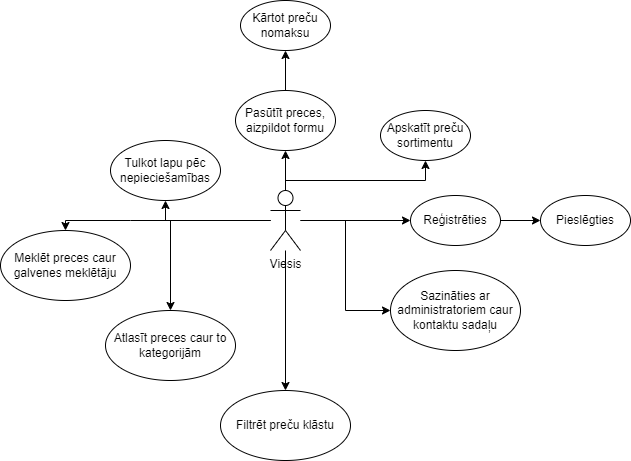
Aktivitāšu diagramma ir UML (Unified Modeling Language) diagrammas veids, kas parāda sistēmas darbības secību, izmantojot kontroli plūsmas konstrukcijas. Tas ietver dažādus elementus, piemēram, darbības, lēmumu mezglus, paralēlo izpildi un notikumu izraisīšanu. Aktivitāšu diagrammas bieži tiek izmantotas, 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).

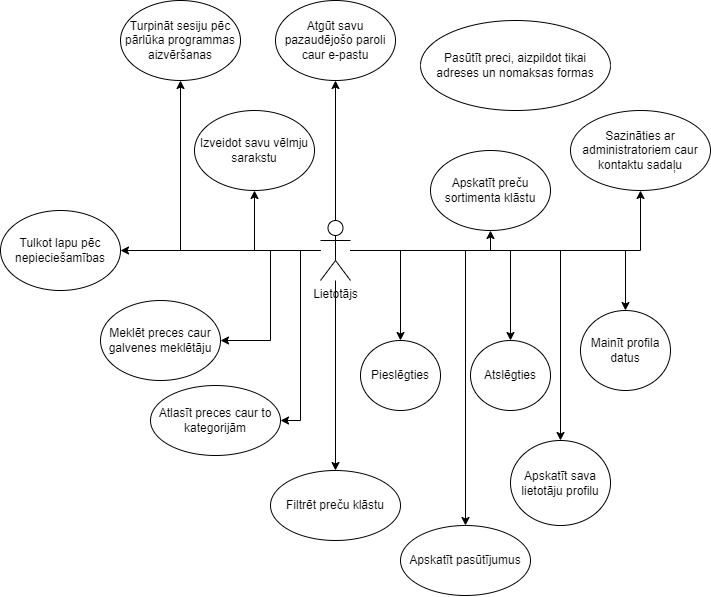


**3. attēls. Lietotāju aktivitāšu diagramma.**

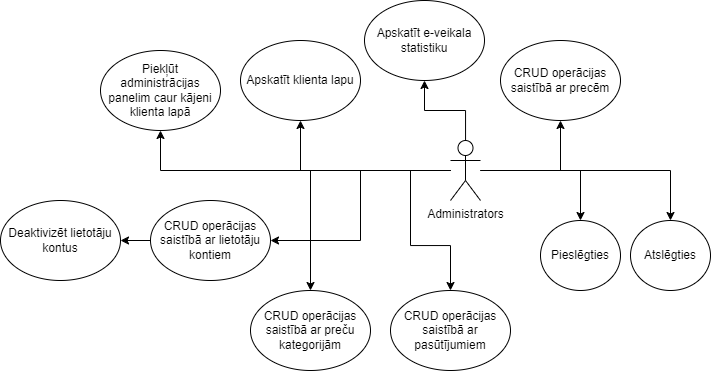
### 4.2.2. Lietojumgadījumu diagramma (Use Case)

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 viesa, 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 viņam izmantot lietotni bez atkārtotas autentifikācijas. Šie soļi ir nepieciešami, lai nodrošinātu drošu un autorizētu piekļuvi lietotāja datiem, saglabājot datu konfidencialitāti un integritāti. Skatīt 7. attēlu.

A diagram of a diagram

Description automatically generated

1. attēls. Google OAuth pieslēgšanas algoritms

### 4.3.2. Paroles atjaunošanas algoritms

A diagram of a diagram

Description automatically generatedParoles 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.

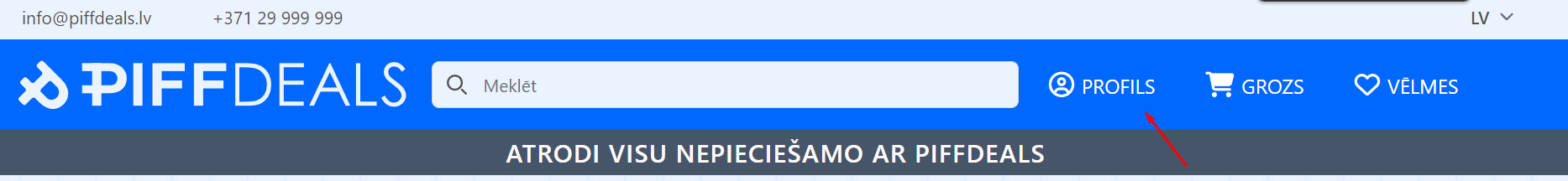
1. Attēls. Paroles atjaunošanas algoritms

# 5. Lietotāju ceļvedis

Ceļvedis nebūs pabeigts, līdz kamēr sistēma nav pilnībā izstrādāta, kā arī lapas dizains mainīsies. Screenshot būs atjaunoti. Šādi, apmēram, būs lietotāju ceļveda rakstīšanas stils.

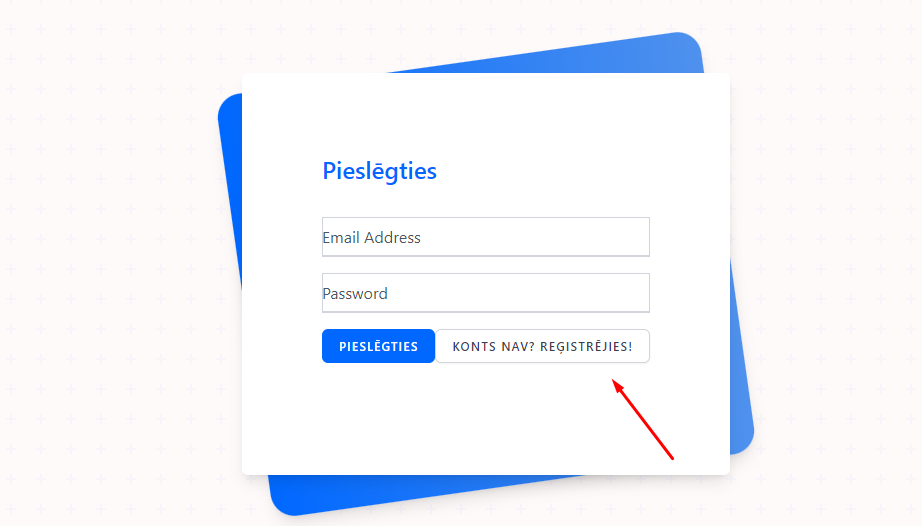
Lietotāja ceļvedis ir svarīgs rīks, lai palīdzētu jauniem un esošiem klientiem efektīvi izmantot e-veikala piedāvātās iespējas un funkcijas. Šāds ceļvedis nodrošina instrukcijas un padomus par to, kā veikt pirkumus, izmantot dažādas meklēšanas un filtrēšanas opcijas, pārvaldīt lietotāja kontu, saprast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

## 5.1. Reģistrācija e-veikalā kā viesis

Reģistrācija e-veikalā nav obligāta, veicot preces pasūtīšanu, toties tā atvieglos klienta dzīvi pērkot citas preces nākotnē, piemēram, nebūs nepieciešamība ievadīt atkārtotus datus par lietotāju. Reģistrācija sistēmā ļaus klientiem veidot savus profilus un pievienot preces vēlmju sarakstā, kā arī apskatīt savu pasūtījumu vēsturi. Skatīt $.attēlu.

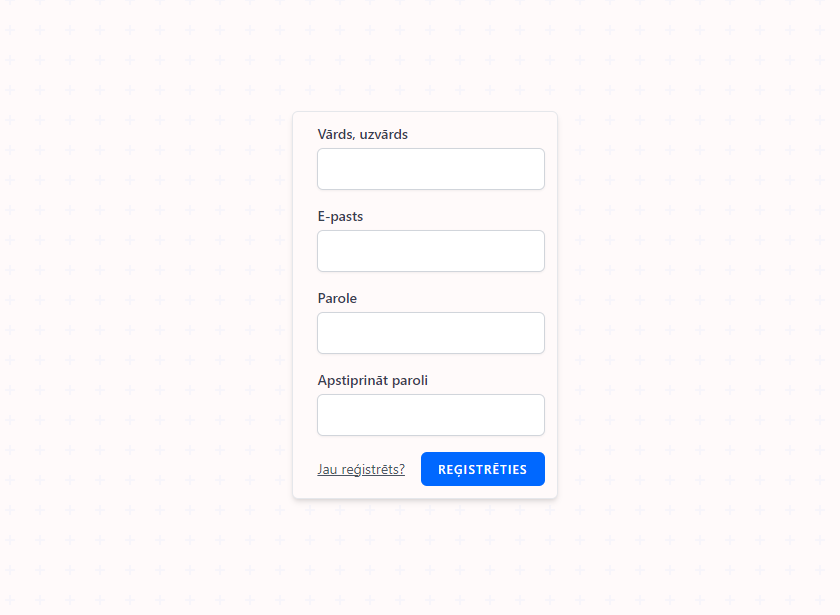
**$.attēls. Bulta uz sadaļu “Profils”.**

E-veikala viesi nav autorizēti apskatīt savu profilu, jo tādu datu vēl nepastāv. Viesiem ir jāreģistrē konts, kurā uzspiešanas uz sadaļas “Profils” gadījumā tiks pieprasīta autorizācija.



**$. Attēls. Autorizācijas logs.**

Viesim jāuzspiež uz attiecīgo pogu un jāreģistrē konts, ievadot nepieciešamos datus formā. Skatīt $. attēlu.



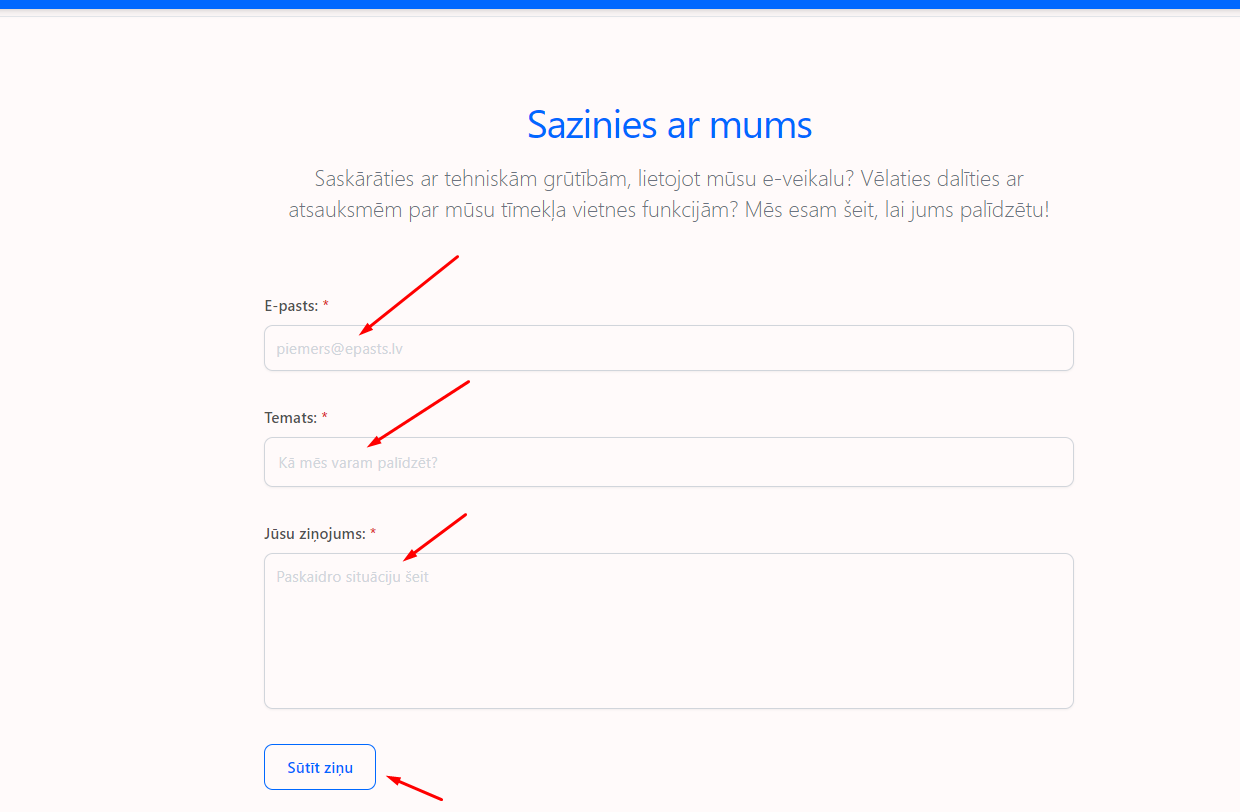
## 5.2. **Sazināšanās ar administratoriem**

Kontaktu sadaļa ir nepieciešama, lai klientu būtu tiesīgi sazināties ar e-veikala administrāciju. Sazināties var jebkurš – viesis un lietotājs, aizpildot sadaļā “Sazinies ar mums”, kura atrodama e-veikala kājenē. Skatīt $. attēlu.



**$. attēls. Kontaktu sadaļas atrašanās vieta.**

Lai nosūtītu ziņu administrācijai, ir jāaizpilda nepieciešamie formas dati un jānospiež poga “Sūtīt ziņu”. Atbilde no administrāciju puses būs vēstulē norādītajā e-pastā. Skatīt $. attēlu.



**$. attēls. Kontaktu sadaļas forma.**

# 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 izvēlējies 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 iespējamas 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 uzliktas lielas slodzes, 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 | PR.01. |
|  |  |  |  |  |  |  |  |
| 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” | PR.SAK.01. |
| 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 | PR.SAK.01. |
|  |  |  |  |  |  |  |  |
| 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 | PR.SAK.02. |
| TP.KL.05 | Profilu sadaļas piekļuve | Jābūt autorizētam un jāuzspiež uz pogas „Profils” | Autorizēts lietotājs ir tiesīgs apmeklēt sadaļa “Profils” | Lietotājm jāautorizējas un jāuzklikšķina uz pogas “Profils” | Lietotājs pieslēdzas sistēmai caur /login | Lietotājs piekļūst sadaļai “Profils” | PR.SAK.02. |
|  |  |  |  |  |  |  |  |
|  | | | | | | | |

**Testpiemēru kopa**

## 6.4. Testēšanas žurnāls

Testēšanas žurnāls satur detalizētu informāciju par veikto testēšanu, tostarp datumi, 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 | 02.06.2024 | TP.KL.05 | Profilu sadaļas piekļuve | Markuss Balodis | Veiksmīgs |  |  |

**Testēšanas žurnāls**

# Secinājumi

Šeit pašlaik nav, ko rakstīt

# 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 $. un $. tabulas, lai gūtu priekšstatu par tās definīcijas nozīmi.

**$. 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 |
| 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 |

**$. tabula**

**Lietoto terminu tabula**

|  |  |
| --- | --- |
| **Termins** | **Apraksts** |
| Back-end | Servera puses daļa, kas apstrādā datu bāzi un lietotāja pieprasījumus |
| Bcrypt | Algoritms drošu paroļu šifrēšanai, plaši izmantots modernās mājaslapās |
| Black box | Programmatūras testēšanas metode |
| Debugging | Atkļūdošanas 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 majaslapu 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 izvietošanai |
| Web hosting platform | Platforma vai pakalpojums, kas sniedz mājaslapu uzturēšanu uz servera |
| Hover | Peles kursora atrašanās 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 sesiju 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 | Definēcijas 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 $. Tabulu.

**$. 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