

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](#_Toc165140412)

[1.Uzdevuma formulējums 5](#_Toc165140413)

[2. Programmatūras prasību specifikācija 6](#_Toc165140414)

[2.1. Produkta perspektīva 6](#_Toc165140415)

[2.2. Sistēmas funkcionālās prasības 6](#_Toc165140416)

[2.2.1. Klienta sadaļas lapa 6](#_Toc165140417)

[2.2.2. Pieejamās preces 11](#_Toc165140418)

[2.2.3. Lietotāja konts 15](#_Toc165140419)

[2.2.4. Administratora informācijas panelis 16](#_Toc165140420)

[2.2.5. Citi 20](#_Toc165140421)

[2.3. Sistēmas nefunkcionālās prasības 21](#_Toc165140422)

[2.4. Gala lietotāja raksturiezīmes 22](#_Toc165140423)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 23](#_Toc165140424)

[3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts 23](#_Toc165140425)

[3.1.1. Back-end puses programmēšanas valodas līdzekļi 23](#_Toc165140426)

[3.1.2. Front-end puses programmēšanas valodas līdzekļi 24](#_Toc165140427)

[3.1.2. Izmantotie rīki 25](#_Toc165140428)

[3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts 26](#_Toc165140429)

[3.3. Līdzekļu un rīku lietojuma pamatojums 27](#_Toc165140430)

[4. Sistēmas modelēšana un projektēšana 28](#_Toc165140431)

[4.1. Sistēmas struktūras modelis 28](#_Toc165140432)

[4.1.1. ER diagramma 28](#_Toc165140433)

[4.2. Funkcionālais un dinamiskais sistēmas modelis 30](#_Toc165140434)

[4.2.1. Aktivitāšu diagramma (Activity) 30](#_Toc165140435)

[4.2.2. Lietojumgadījumu diagramma (Use Case) 32](#_Toc165140436)

[4.3. Sistēmas moduļu apraksts un algoritmu shēmas 34](#_Toc165140437)

[5. Lietotāju ceļvedis 35](#_Toc165140438)

[5.1. Reģistrācija e-veikalā kā viesis 35](#_Toc165140439)

[5.2. Sazināšanās ar administratoriem 36](#_Toc165140440)

[6. Testēšanas dokumentācija 38](#_Toc165140441)

[6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums 38](#_Toc165140442)

[6.2. Testpiemēru kopa 38](#_Toc165140443)

[6.3. Testēšanas žurnāls 38](#_Toc165140444)

[7. Secinājumi 39](#_Toc165140445)

[8. Lietoto terminu un saīsinājumu skaidrojumi 40](#_Toc165140446)

[9. Literatūras un informācijas avotu saraksts 42](#_Toc165140447)

[Pielikums 43](#_Toc165140448)

# 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**

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. Jaunāko preču ielāde lapas sākumlapā**

Mērķis:

Ielādēt jaunākās, pieejamās preces e-veikala sākumlapā

Ievaddati:

Lietotājam vai administratoram ir jāapmeklē e-veikala sākumlapai un preču administratoram ir jāizveido vismaz viena prece un jāiespējo tā, mainot tās statusu uz “Pieejams”

Apstrāde:

Funkcija pārbauda vai lietotājs vai administrators pašlaik apmeklē e-veikala sākumlapu un administrators ir izveidojis vismaz vienu preci un iespējojis to

Izvaddati:

E-veikala lietotājam tiek ielādētas jaunākas preces sākumlapā

**P.3. Preču kategoriju ielāde sākumlapā**

Mērķis:

Preču kategoriju ielādi sākumlapā

Ievaddati:

E-veikala lietotājam ir jāapmeklē sākumlapa un administratoram ir jāizveido vismaz viena prece, un tā ir pieejama

Apstrāde:

Sākumlapas ielādēšanas procesā tiek iegūtu dati par precēm no datu bāzes, kas atlasīs konkrētas preces kategoriju

Izvaddati:

Funkcija izvadīs preču kategoriju vienu reizi sākumlapā

**P.4. Preču apakškategoriju ielāde sākumlapā**

Mērķis:

Uzrādīt preču kategorijas apakškategorijas (ja tās ir), tiklīdz lietotājs ir uzlicis peles kursoru *(hover)* uz jebkādu preču kategoriju

Ievaddati:

E-veikala lietotājam ir jāapmeklē sākumlapa un administratoram ir jāizveido vismaz viena prece, kurai sastāv viena kategorija un apakškategorija, kā arī jābūt pieejamai

Apstrāde:

Funkcija pārbauda preces informāciju un vai lietotājs ir uzlicis kursoru uz vienu no preču kategorijām

Izvaddati:

Apstiprinājuma gadījumā, attiecošais HTML elements tiek parādīts ekrānā un uzrāda visas kategoriju apakškategorijas

**P.5. Peles kreisais klikšķis uz preci**

Mērķis:

Klikšķis uz preci, ielādēs lapu ar tās preces detalizēto informāciju

Ievaddati:

1. E-veikala klientam jāapmeklē lapas, kurā piedāvā preces un ar peles kreiso klikšķa palīdzību, uzspiež uz preci;
2. Administratoram jāizveido preces, kurām statuss ir “Pieejams”;

Apstrāde:

Funkcija pārbauda lietotāja *onClick* informāciju un ielādē citu lapu

Izvaddati:

Tiek ielādēta cita lapa, kurā ir tās attieciešošās preces saturs

**P.6. Preču filtrs**

Mērķis:

Sniegt lietotājam iespēju filtrēt pieejamās preces caur dažādiem veidiem

Ievaddati:

1. Lietotājam jāizvēlas preču apakškategorija;
2. Administratoram jāizveido vismaz 10, pieejamās preces;
3. Lietotājam jāorientējas filtru HTML elementu sadaļā un jāizvēlas vienu no veidiem, kā filtrēt preces

Apstrāde:

Lietotājs izvēlas veidu kā filtrēt preces un funkcija izvēlēto veidu aprēķina un pēc tam pārlādē lapu

Izvaddati:

Lapu pārlādēšanas gadījumā, uz lapas tiek uzrādītas preces, kuras ietilpst filtrēšanas diapzonā

**P.9. Preču filtrēšanas ar atslēgasvārdiem**

Mērķis:

Nodrošināt iespēju filtrēt pieejamās preces caur atslēgasvārdiem

Ievaddati:

1. Lietotājam jāizvēlas preču apakškategorija;
2. Administratoram jāizveido vismaz 10, pieejamās preces;

Apstrāde:

Funkcija aprēķina ievadīto informāciju ievades laukā un pēc tam ar atslēgasvārdiem filtrē preces, kurā atslēgasvārdi iekļaujas preces nosaukumā

Izvaddati:

Funkciju aprēķinu beigās tiek pārlādēta lapa, kurā tiek atlasītas preces, kuru nosaukumā sastāv atslēgasvārds

**P.10. Preču filtrēšana, izvēloties cenu diapzonu**

Mērķis:

Nodrošināt iespēju filtrēt pieejamās preces, izvēloties cenu diapzonu

Ievaddati:

1. Lietotājam jāizvēlas preču apakškategorija;
2. Administratoram jāizveido vismaz 10, pieejamās preces;

Apstrāde:

Funkcija pārbauda un aprēķina cenu diapzonu un pēc tam filtrē preces, kuras iekļaujas diapzonā

Izvaddati:

Funkciju aprēķinu beigās tiek pārlādēta lapa, kurā tiek atlasītas preces, kuras iekļaujas cenu diapzonā

**P.11. Preču filtrēšana, norādot konkrētu ražotāju**

Mērķis:

Nodrošināt iespēju filtrēt pieejamās preces, norādot ražotāju

Ievaddati:

1. Lietotājam jāizvēlas preču apakškategorija;
2. Administratoram jāizveido vismaz 10 pieejamās preces;

Apstrāde:

Funkcija pārbauda un aprēķina norādīto preču ražotāju un pēc tam filtrē preces, kurām ir viens ražotājs

Izvaddati:

Funkciju aprēķinu beigās tiek pārlādēta lapa, kurā tiek atlasītas preces, kuras iekļaujas zem viena ražotāja

**P.12. Preču kārtošana dilstoši pēc to cenas**

Mērķis:

Sniegt iespēju lietotājam kārot preces dilstoši pēc to cenas

Ievaddati:

1. Lietotājam jāizvēlas preču apakškategorija;
2. Administratoram jāizveido vismaz 10, pieejamās preces;

Apstrāde:

Lietojot HTML Select, lietotājs orientējoties izvēlas opciju kārtot preces dilstoši. Funkcija pārbauda vai opcija tiek izvēlēta un atlasa visas preces dilstoši pēc to cenas

Izvaddati:

Funkciju aprēķinu beigās tiek pārlādēta lapa, kurā tiek atlasītas preces, kurām kārtojums ir dilstošs pēc to cenas

**P.13. Preču kārtošana augoši pēc to cenas**

Mērķis:

Sniegt iespēju lietotājam kārot preces augoši pēc to cenas

Ievaddati:

1. Lietotājam jāizvēlas preču apakškategorija;
2. Administratoram jāizveido vismaz 10, pieejamās preces;

Apstrāde:

Lietojot HTML Select, lietotājs orientējoties izvēlas opciju kārtot preces augoši. Funkcija pārbauda vai opcija tiek izvēlēta un atlasa visas preces augoši pēc to cenas

Izvaddati:

Funkciju aprēķinu beigās tiek pārlādēta lapa, kurā tiek atlasītas preces, kurām kārtojums ir augošs pēc to cenas

**P.14. Navigācijas joslas meklēšanas lauka orientācija**

Mērķis:

Sniegt iespēju lietotājam atrast sev vēlamo preci, ievadot meklēšanas joslā tā atslēgas vārdu, kas atrodas preču nosaukumā

Ievaddati:

Administratoram ir jāizveido preces, kuras ir pieejamas

Apstrāde:

Ievadot atslēgasvārdu meklēšanas joslā, tiek iespējots meklēšanas rezultātu elements

Izvaddati:

Meklēšanu rezultātu elementu iespējošanas gadījumā, tiek uzrādīti dažādi preču varianti, kuriem nosaukumā sastāv ievadošais atslēgasvārds. Atslēgasvārds tiek izcelts pēc izstrādātāju ieskata

**P.15. Navigācijas joslas meklēšanas lauka orientācija**

Mērķis:

Sniegt iespēju lietotājam atrast sev vēlamo preci, ievadot meklēšanas joslā tā atslēgas vārdu, kas atrodas preču nosaukumā

Ievaddati:

Administratoram ir jāizveido preces, kuras ir pieejamas

Apstrāde:

Ievadot atslēgasvārdu meklēšanas joslā, tiek iespējots meklēšanas rezultātu elements

Izvaddati:

Meklēšanu rezultātu elementu iespējošanas gadījumā, tiek uzrādīti dažādi preču varianti, kuriem nosaukumā sastāv ievadošais atslēgasvārds. Atslēgasvārds tiek izcelts pēc izstrādātāju ieskata

### 2.2.2. Pieejamās preces

**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. Groza informācijas nolasīšana un saglabāšana**

Mērķis:

Lietotājs var sekmīgi ievietot izvēlēto preci savā grozā

Ievaddati:

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

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.18. Groza informācijas nolasīšana un saglabāšana e-veikala reģistrētam lietotājam caur localStorage**

Mērķis:

E-veikala autorizēts lietotājs var saglabāt un nolasīt groza datus caur pārlūkprogrammas *localStorage*

Ievaddati:

1. Administratoriem ir jāizveido jaunas, pieejamas preces;
2. Lietotājam jābut autorizētam;
3. Lietotājam jāuzspiež uz pogas “Ievietot grozā”;

Apstrāde:

Nospiežot pogu “Ievietot grozā”, lietotāja izvēlētā preces identifikators tiek ievietots *localStorage* kā *JSON* elements

Izvaddati:

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

**P.19. Groza informācijas nolasīšana un saglabāšana, mainot preces daudzumu**

Mērķis:

Lietotājs var sekmīgi ievietot izvēlēto preci savā grozā, mainot tās daudzumu

Ievaddati:

Administratoram ir jāizveido preces, kuras ir pieejamas un lietotājam ir jāapmeklē konkrētās preces lapa, lai nospiestu pogu “Ievietot grozā”, un jāizvēlas konkrēts skaits konkrētai precei

Apstrāde:

Funkcija pārbauda vai preču daudzums ir modificēts, jo pēc noklusējuma skaits ir 1 un ir nospiesta poga “Ievietot grozā”

Izvaddati:

Tiek izvadīts sekmīgs paziņojums un lietotājs var izvēlēties turpināt iepirkšanos vai uzreiz apskatīt grozu. Ja lietotājs apskata grozu, tad ir redzama ievietotā prece ar tās mainīto daudzumu

**P.20. Preču vairuma skaita maksimālā vērtība pirms ievietošanas grozā**

Mērķis:

Lietotājs var mainīt preces daudzumu pirms ievietošanas grozā līdz noteiktam skaitlim

Ievaddati:

Administratoram ir jāizveido preces, kuras ir pieejamas un lietotājam ir jāapmeklē konkrētās preces lapa, kurā var mainīt preces daudzumu pirms ievietošanas grozā

Apstrāde:

Funkcija dinamiski iestatīs maksimālo vērtību pēc preču daudzuma no datubāzes

Izvaddati:

Preces daudzuma maksimālā vērtība ir atkarīga no preču daudzuma, kas ir noliktavā jeb datubāzē

**P.21. Iepirkuma groza apskate**

Mērķis:

Sniegt iespēju e-veikala reģistrētam lietotājam apskatīt savu iepirkuma grozu

Ievaddati:

Administratoram ir jāizveido preces, kuras ir pieejamas un lietotājam ir jāapmeklē konkrētās preces lapa un jānospiež poga “Ievietot grozā”

Apstrāde:

Funkcija ielādē visas lietotāja saglabātās preces, kuras ir ievietotas grozā un sniedz iespēju lietotājam veikt samaksu, ja visas preces ir pieejamas

Izvaddati:

Lietotājs apmeklē sava groza lapu, kurā ir visas izvēlētās preces

**P.22. Preču vispārējā pieejamība**

Mērķis:

Aprakstīt preces pieejamību

Ievaddati:

Administratoram ir jāizveido preces – tās var būt gan pieejamas, gan nepieejamas. Preču statusi ir šādi: pieejams, izpārdots, nepieejams

Apstrāde:

Funkcija sākumā pārbauda preces daudzumu noliktavā un ja ir 0, tad preces statuss ir izpārdots, citu iemeslu dēļ prece ir nepieejama vai pieejama, proti, ja prece ir izpārdota vai nepieejama un tiek pievienota grozam, tad turpmākos soļus nevar veikt sakarā ar apmaksāšanu un piegādi

Izvaddati:

E-veikala klients uzzina preces pieejamību caur konkrētās preces lapu

**P.23. Preču vēlmju saraksta apskate**

Mērķis:

Aprakstīt savu preču vēlmju sarakstu, kamēr lietotājs ir autorizējies

Ievaddati:

Administratoram ir jāizveido preces un e-veikala klientam ir jāapmeklē konkrētās preces lapa un jānospiež poga “Pievienot vēlmju sarakstā”

Apstrāde:

Vēlmju saraksta lapā tiks uzrādīti e-veikala klienta saglabātās preces, neatkarīgi no tā statusa

Izvaddati:

E-veikala klients uzzina preces pieejamību caur konkrētās preces lapu

**P.24. Preču vēlmju saraksta papildināšana ar precēm**

Mērķis:

Autorizēts lietotājs var pievienot preces savam vēlmju sarakstam

Ievaddati:

Administratoram ir jāizveido preces un e-veikala autorizētam klientam ir jāapmeklē konkrētās preces lapa un jānospiež poga “Pievienot vēlmju sarakstā”

Apstrāde:

Funkcija saglabā preces identifikatoru jeb SKU, kurā to pēc tam ievieto lietotāja vēlmju sarakstā caur *localStorage*

Izvaddati:

Tiek izvadīts paziņojums par sekmīgu preču pievienošanu vēlmju sarakstā

### 2.2.3. Lietotāja konts

**P.25. 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.26. 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.26. 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.27. 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

### 2.2.4. Administratora informācijas panelis

**P.28. Pasūtījumu lapas apskate**

Mērķis:

Sniegt iespēju administratoram/piegādātājam apskatīt klienta pasūtījumus

Ievaddati:

Administratoram jāizveido preces, kurus varētu interesēt e-veikala klients. Klients tās preces ievieto grozā un veic apmaksāšanu

Apstrāde:

Funkcija pārbauda vai apmaksa ir veikta un atlasa tās pasūtījuma informāciju un uzrāda to tabulas veidā sadaļā “Pasūtījumi”

Izvaddati:

Administrators/piegādātājs var apskatīt klientu pasūtījumus

**P.29. Pēc nepieciešamības, administrators var veikt *CRUD* funkcijas sakarā ar pasūtījumiem**

Mērķis:

Citu iemeslu dēļ, kā nepareizu pasūtījumu izveidi, administratoram sniegt iespēju veikt jebkādas *CRUD* funkcijas sakarā ar klienta preču pasūtījumiem

Ievaddati:

E-veikala klientam jāveic preču pasūtījums un jānokārto apmaksāšana

Apstrāde:

Funkcijas atļauj administratoram veikt pasūtījumu rediģēšanu, dzēšanu un lasīt

Izvaddati:

Administrators, pēc nepieciešamības, var manipulēt klienta pasūtījumus.

**P.30. Lietotāju kontu apskate**

Mērķis:

Administratori var apmeklēt sadaļu “Lietotāji” un apskatīt kontus, izņemot šifrētās paroles un *tokens*

Ievaddati:

Administratoram jābūt autorizētam, lai apmeklētu sadaļu “Lietotāji”

Apstrāde:

Funkcija ielādē visus lietotājus no datubāzes, izņemot laukus kā paroles un *tokens*

Izvaddati:

Apmeklējot sadaļu “Lietotāji”, administrators var apskatīt lietotāju kontus tabulas veidā

**P.31. Pēc nepieciešamības, administrators var veikt *CRUD* funkcijas sakarā ar lietotāju kontiem**

Mērķis:

Citu iemeslu dēļ, sniegt administratoram iespēju manipulēt ar lietotāja konta datiem

Ievaddati:

Administratoram jābūt autorizētam un jāapmeklē sadaļa “Lietotāji”

Apstrāde:

Funkcijas atļauj administratoram veikt izveidošanu, rediģēšanu, lasīšanu un dzēšanu sakarā ar lietotāju kontiem

Izvaddati:

Administrators, pēc nepieciešamības, var manipulēt e-veikala lietotāju kontus caur tabulas ietvariem

**P.32. Jaunu preču pievienošana**

Mērķis:

Sniegt administratoram iespēju pievienot jaunas preces e-veikala sortimentā

Ievaddati:

Administratoram jābūt autorizētam un jāapmeklē sadaļa “Preces”, un ar peles kreiso klikšķi, jānospiež poga “Pieviento jaunu preci”

Apstrāde:

Funkcija pārbauda vai forma tiek aizpildīta un pēc pogas nospiešas, pievieno preci datubāzē, kuram statuss pēc noklusējuma ir neaktīvs

Izvaddati:

Tiek izvadīts paziņojums par sekmīga preces pievienošanu

**P.33. Preču rediģēšana**

Mērķis:

Sniegt administratoram iespēju rediģēt preces e-veikala sortimentā

Ievaddati:

Administratoram jābūt autorizētam un jāapmeklē sadaļa “Preces”, un ar peles kreiso klikšķi, jānospiež poga “Rediģēt preci”

Apstrāde:

Funkcija pārbauda vai prece ir rediģēta un iespējo jeb ieslēdz saglabāšanas pogu, kas ļauj lietotājam saglabāt izmaiņas

Izvaddati:

Rediģēšanas logs aizveras un tabulas vai lapa tiek pārlādēta ar jaunām izmaiņām

**P.34. Preces rediģēšanas logā jeb *modal* ievades laukos tiek ielādēta preces informācija**

Mērķis:

Preces rediģēšanas procesā, atverot rediģēšanas logu, ievades lauki tiek aizpildīti ar izvēlēto preces informāciju

Ievaddati:

Administratoram jābūt autorizētam un jāapmeklē sadaļa “Preces” un jānospiež poga “Rediģēt preci”

Apstrāde:

Funkcija *modal* logā, ievades laukos ielādē konkrētās preces datus

Izvaddati:

Ievades lauki nav tukši, bet piepildīti ar preces informāciju

**P.35. Sadaļā “Preces” lappušu numerācija funkcionalitāte**

Mērķis:

Sniegt iespēju administratoram orientēties sadaļā “Preces” caur lappušu numerācijām, piemēram, nevis atlasītu visus preču datus vienā lapā, bet sadalīt tās pa vairākām lapām caur lappušu numerācijām

Ievaddati:

Administratoram jābūt autorizētam un jāapmeklē sadaļa “Preces”. Lai lappušu numerācijas funkcionalitāte sekmīgi strādātu, tam ir jāizveido vismaz 20 jaunas, pieejamas preces

Apstrāde:

Funkcija veido lappušu numerācijas atkarīgi no preču lauka skaita sadaļā “Preces”

Izvaddati:

Lappušu numerācijas skaits pakāpinās vai samazinās atkarīgi no preču lauka skaita un, uzklikšķinot uz jebkuru lappušu numerācijas skaitli, tas pārlādēs lapu ar citām precēm

**P.36. Preču noklusējuma attēls**

Mērķis:

Ja preču attēls netiek sekmīgi ielādēts no servera, tad tās alternatīvais attēls būs e-veikala *favicon* jeb ikona

Ievaddati:

Iestatīt nepareizu jeb salauztu attēlu kā preces attēlu, kad pievieno jaunu preci sadaļā “Preces”

Apstrāde:

Funkcija iestatīs preces noklusējamo attēlu, ja no servera preces attēls nav veiksmīgi ielādējies

Izvaddati:

Preces attēls ir e-veikala *favicon* jeb dizaina ikona

### 2.2.5. Citi

**P.37. 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.38. 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.39. 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.40. 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.

# 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.

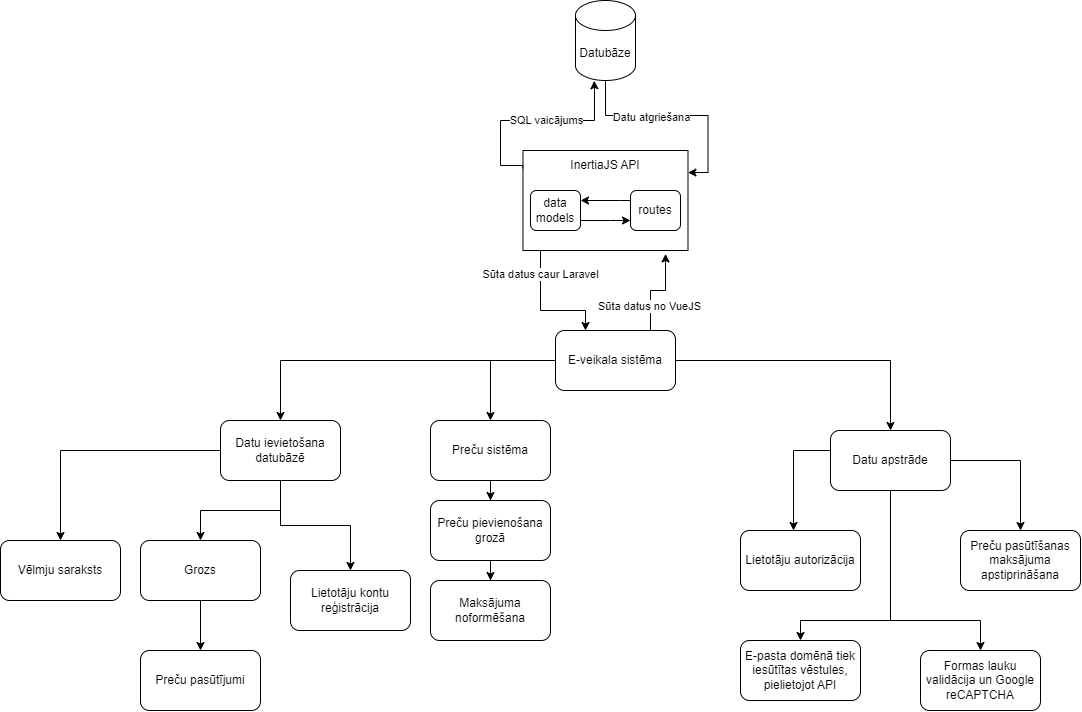
Laravel un Vue labāk sadarbojas nekā Laravel un React, piemēram, Laravel piedāvā plašāk sarakstītu dokumentāciju priekš Vue, nevis React. Tika apdomāta stratēģija, kā panākt front-end un back-end komunikāciju, tai vietā nāk InertiaJS, kas palīdz komunicēt starp abām pusēm un padot tālāk precīzu informāciju. 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

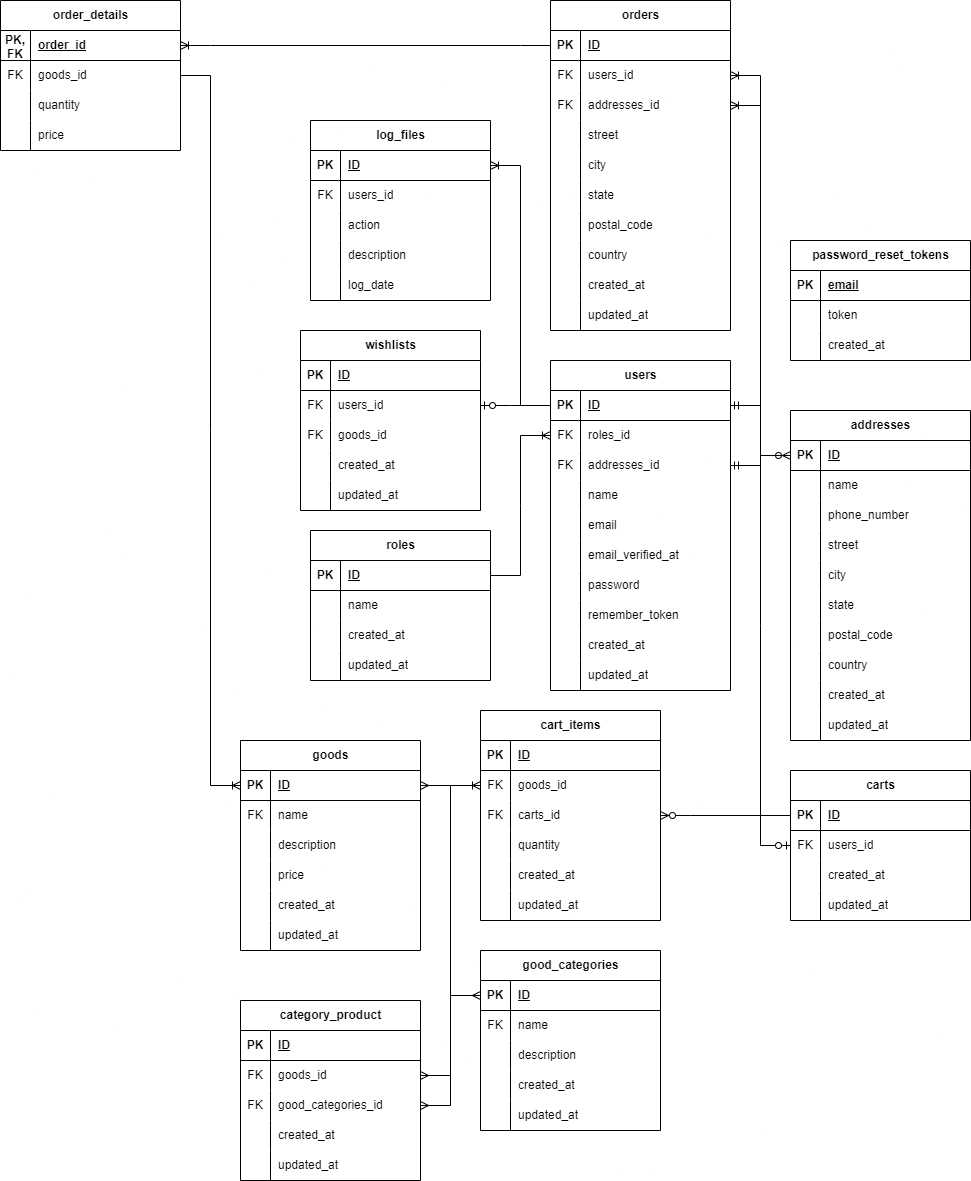
Sistēmas struktūras modelis ir attiecināms pret sistēmas komponenšu un to attiecību reprezentāciju. Tās mērķis ir sniegt palīdzību izstrādātājiem un sistēmas analītiķiem izprast, kā sistēma tiek izstrādāta un organizēta, tā sniedz pamatu sistēmas arhitektūrai, izveidei un analīzei. Modelis attēlo datu 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**

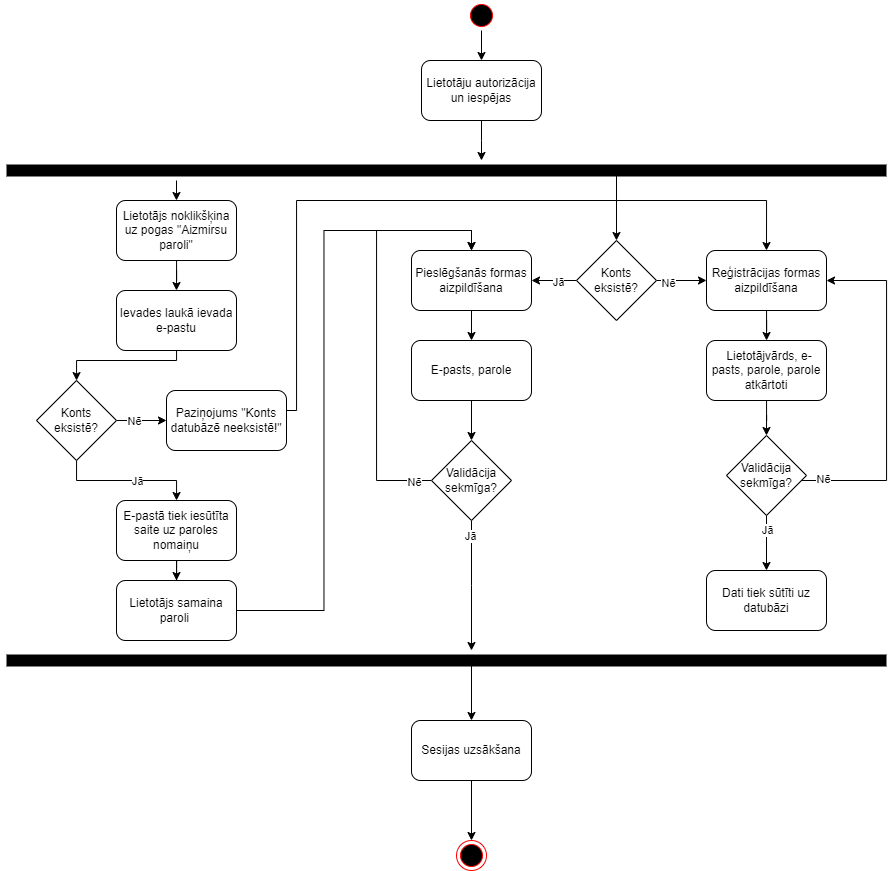
## 4.2. Funkcionālais un dinamiskais sistēmas modelis

aaaa

### 4.2.1. Aktivitāšu diagramma (Activity)

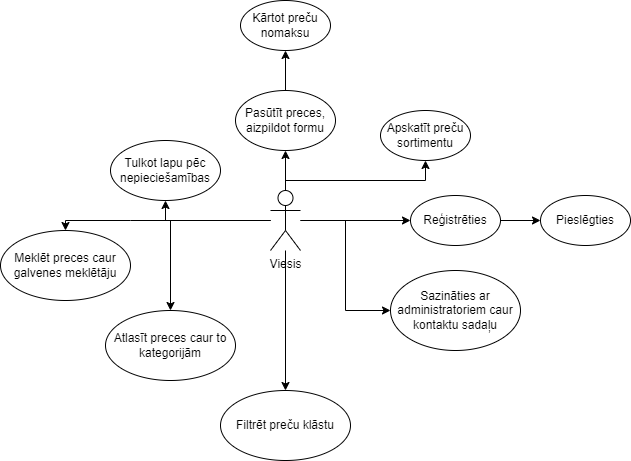
Aktivitāšu diagramma ir UML (Unified Modeling Language) diagrammas veids, kas parāda sistēmas darbības secību, izmantojot kontroli plūsmas konstrukcijas. Tas ietver dažādus elementus, piemēram, darbības, lēmumu mezglus, paralēlo izpildi un notikumu izraisīšanu. Aktivitāšu diagrammas bieži tiek izmantotas,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

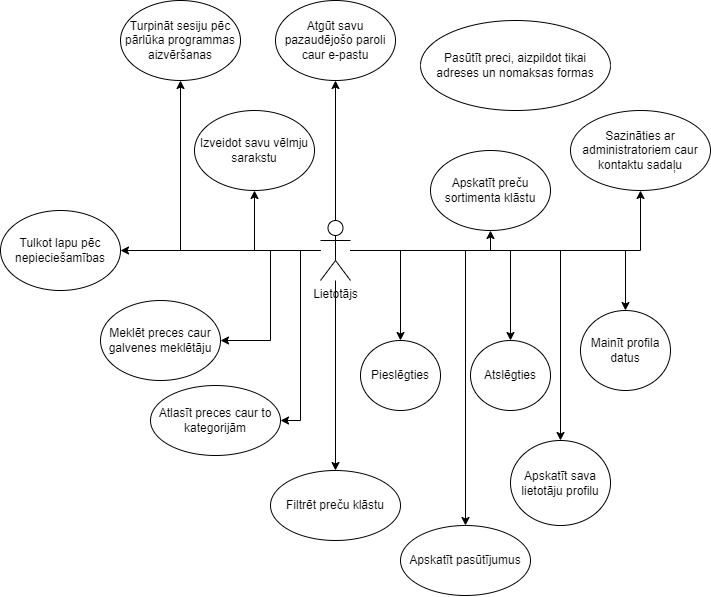


**3. attēls. Lietotāju aktivitāšu diagramma.**

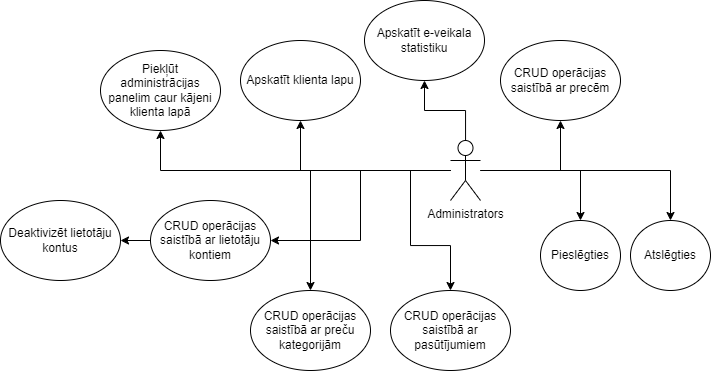
### 4.2.2. Lietojumgadījumu diagramma (Use Case)

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

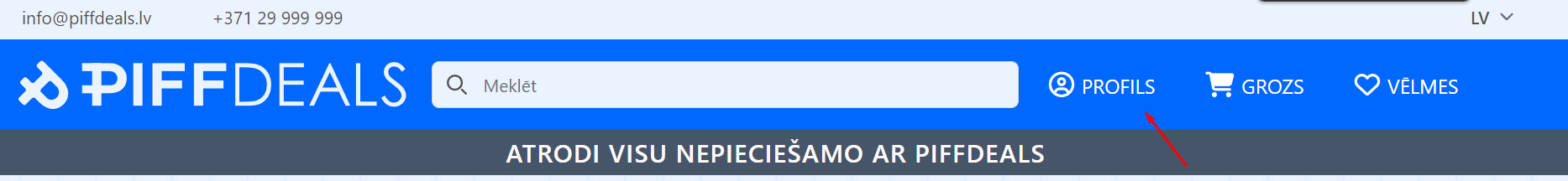
aaaa

# 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.

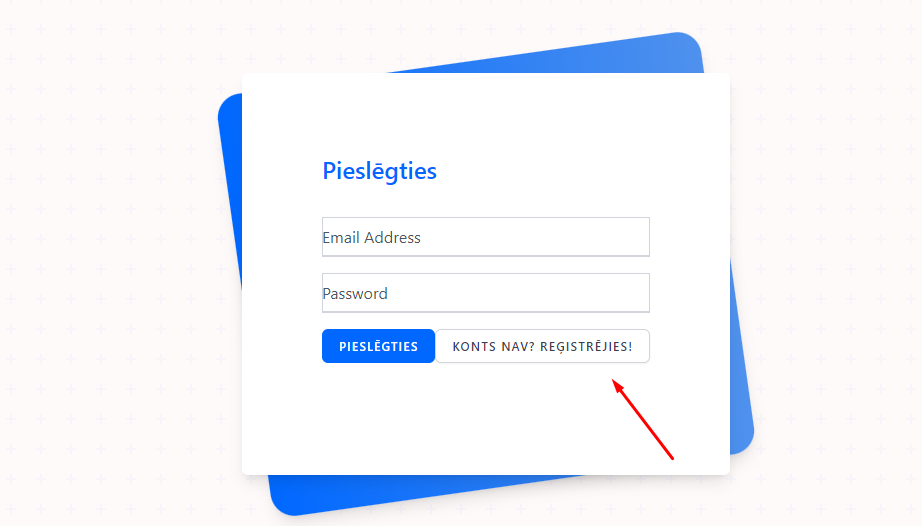
Lietotāja ceļvedis ir svarīgs rīks, lai palīdzētu jauniem un esošiem klientiem efektīvi izmantot e-veikala piedāvātās iespējas un funkcijas. Šāds ceļvedis nodrošina instrukcijas un padomus par to, kā veikt pirkumus, izmantot dažādas meklēšanas un filtrēšanas opcijas, pārvaldīt lietotāja kontu, saprast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

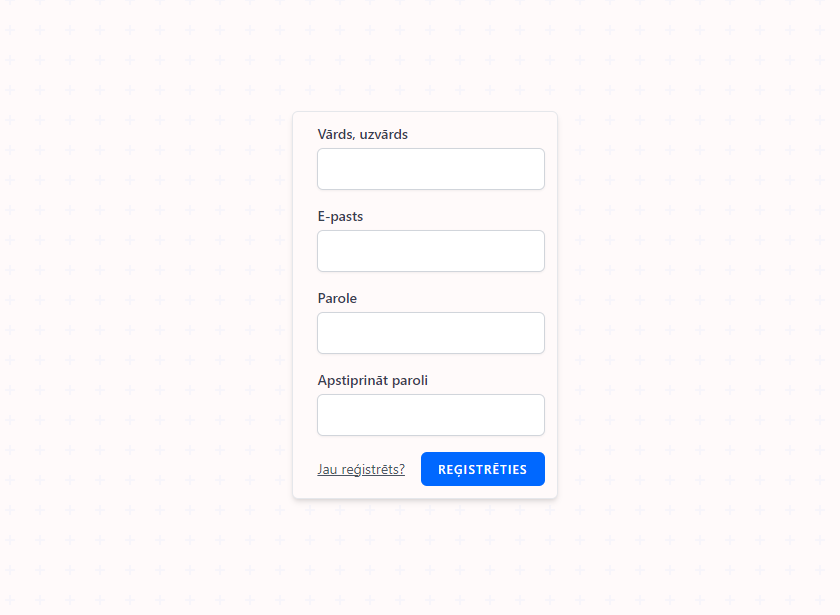
**$.attēls. Bulta uz sadaļu “Profils”.**

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.



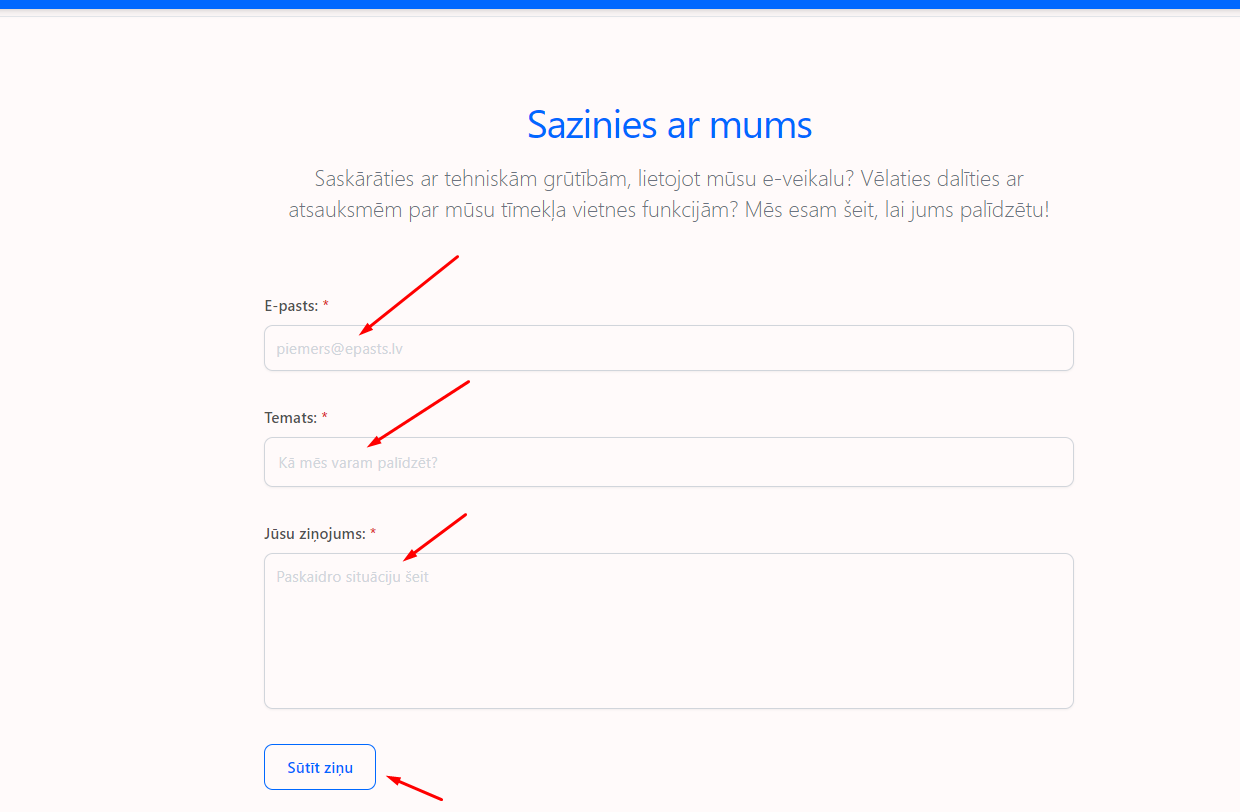
## 5.2. **Sazināšanās ar administratoriem**

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.

## 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

# 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