

Liepājas Valsts tehnikums

**Internetveikals “Sparkly Dream”**

Kvalifikācijas eksāmena praktiskās daļas tehniskā dokumentācija

Izglītības programma **33484011 Programmēšana**

Personālā kvalifikācija **Programmēšanas tehniķis**

Darba autors(-e): ………………………………………………

/vārds, uzvārds, paraksts/

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

Liepāja 202\_\_

# Saturs

[Saturs 3](#_Toc198478135)

[Ievads 6](#_Toc198478136)

[1. Uzdevuma formulējums 7](#_Toc198478137)

[2. Programmatūras prasību specifikācija 9](#_Toc198478138)

[2.1. Produkta perspektīva 9](#_Toc198478139)

[2.2. Sistēmas funkcionālās prasības 9](#_Toc198478140)

[2.3. Sistēmas nefunkcionālās prasības 24](#_Toc198478186)

[2.4. Gala lietotāja raksturiezīmes 24](#_Toc198478187)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 26](#_Toc198478188)

[3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts 26](#_Toc198478189)

[3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts 28](#_Toc198478190)

[4. Sistēmas modelēšana un projektēšana 30](#_Toc198478191)

[4.1. Sistēmas struktūras modelis 30](#_Toc198478192)

[4.1.1. Izvietojuma (Deployment) diagramma 30](#_Toc198478193)

[4.1.2. ER (Entity Relationship) diagramma 31](#_Toc198478194)

[4.2. Funkcionālais un dinamiskais sistēmas modelis 32](#_Toc198478195)

[4.2.1. Aktivitāšu (Activity) diagramma 32](#_Toc198478196)

[4.2.2. Lietojumgadījumu (Use Case) diagramma 33](#_Toc198478197)

[4.2.3. Stāvokļu (State) diagramma (Produkta pievienōšana grozā) 34](#_Toc198478198)

[4.3. Datu struktūru apraksts 35](#_Toc198478199)

[5. Lietotāju ceļvedis 37](#_Toc198478200)

[5.1. Klienta ceļvedis 37](#_Toc198478201)

[5.1.1. Ielogošanās 37](#_Toc198478202)

[5.1.2. Reģistrācija 37](#_Toc198478203)

[5.1.3. Izlogošanās 38](#_Toc198478204)

[5.1.4. Produkta pievienošana grozā 38](#_Toc198478205)

[5.1.5. Produkta daudzuma rediģēšana grozā 39](#_Toc198478206)

[5.1.5. Produkta dzēšana no groza 39](#_Toc198478207)

[5.1.6. Groza tīrīšana 39](#_Toc198478208)

[5.1.7. Pasūtījuma noformēšana 39](#_Toc198478209)

[5.1.8. Personas datu rediģēšana 40](#_Toc198478210)

[5.1.9. Paroles maiņa vai konta dzēšana 40](#_Toc198478211)

[5.1.10. Saziņa ar organizāciju 40](#_Toc198478212)

[6. Testēšanas dokumentācija 41](#_Toc198478213)

[6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums 41](#_Toc198478214)

[6.2. Alternatīvās testēšanas metodes un rīki 41](#_Toc198478215)

[6.3. Testpiemēru kopa 42](#_Toc198478216)

[6.4. Testēšanas žurnāls 48](#_Toc198478218)

[Secinājumi 50](#_Toc198478219)

[7. Lietoto saīsinājumu un terminu skaidrojums 51](#_Toc198478220)

[Literatūras un informācijas avotu saraksts 52](#_Toc198478221)

[Pielikumi 53](#_Toc198478222)

# Ievads

Šis projekts ir paredzēts unikāla internetveikala "Sparkly Dream" izstrādei, kura ideja balstīta darba autora ģimenes uzņēmuma pieredzē. Uzņēmums ir izveidojies kā neliels, bet radošs darbnīcas projekts, kas specializējas unikālu Ziemassvētku eglīšu rotājumu izstrādē. Roku darbs ir uzņēmuma pamatvērtība, piešķirot rotājumiem īpašu izskatu un augstu kvalitāti. Līdz šim uzņēmuma produkcija tika piedāvāta tikai lokāli, taču pieaugošā interese par personalizētiem un augstvērtīgiem rotājumiem ir veicinājusi nepieciešamību paplašināt darbību.

Dokumentā ir aprakstītas šādas sadaļas: uzdevuma formulējums, programmatūras prasību specifikācija, izstrādes līdzekļu un rīku apraksts ar izvēles pamatojumu, sistēmas modelēšana un projektēšana, lietotāju ceļvedis, testēšanas dokumentācija, secinājumi, lietoto saīsinājumu un terminu skaidrojums, kā arī literatūras un informācijas avotu saraksts.

“Sparkly Dream” ir vairāk nekā tikai veikals – tā ir vieta, kur ikviens var radīt kaut ko īpašu un unikālu saviem svētkiem. Viena no svarīgākajām platformas funkcijām ir iespēja klientiem pielāgot savus rotājumus pēc savas gaumes un vēlmēm. Klienti var izvēlēties pamatkrāsu, kas veidos rotājuma galveno toni, kā arī pielāgot rotājuma malas figūru – ārējo formu vai dekoratīvu akcentu. Šīs figūras var būt, piemēram, eņģeļi, putni vai citi simboli, kas piešķir rotājumam individuālu raksturu. Ka arī klienti var izvēlēties arī rotājuma formu, kas noteiks, kādā stilā vai veidā rotājums izskatīsies – apaļa, medaljona, lāsteka vai pat lietussargs, lai atbilstu konkrētajam svētku tematam. Šī pielāgošanas iespēja ļauj radīt personalizētus un neaizmirstamus dekorus, kas atspoguļo klienta gaumi un svētku noskaņu. "Sparkly Dream" ir radīts, lai katra ideja pārvērstos par mirdzošu realitāti! Tā ir aktuāla ikvienam, kurš meklē ne tikai rotājumu, bet arī radošu pieredzi un iespēju radīt kaut ko īpašu.

Projekta mērķis ir attīstīt ģimenes uzņēmumu, izveidojot ērtu un lietotājam draudzīgu internetveikalu "Sparkly Dream". Mājaslapa nodrošinās klientiem vienkāršu veidu, kā apskatīt un pasūtīt ar rokām darinātus Ziemassvētku eglīšu rotājumus. Tiks piedāvāta arī iespēja personalizēt rotājumus, tādējādi apmierinot individuālās klientu vēlmes. Šis projekts palīdzēs sasniegt plašāku auditoriju, palielinās uzņēmuma atpazīstamību un veicinās pārdošanas apjomus, radot modernu un pieejamu platformu unikālu produktu iegādei.

Līdztekus biznesa attīstībai projekts palīdzēs autoram paplašināt savas zināšanas programmēšanā, iegūstot praktisku pieredzi un pielietojot mūsdienīgas tehnoloģijas programmatūras izstrādē. Tādējādi tas ne tikai stiprinās uzņēmuma konkurētspēju, bet arī sniegs būtisku profesionālo izaugsmi projektā iesaistītajām personām, radot vērtību gan pašam autoram, gan lietotājiem, gan visai uzņēmējdarbības videi.

# Uzdevuma formulējums

Projekta ietvaros tiks izstrādāts internetveikals "Sparkly Dream", kura mērķis ir nodrošināt klientiem ērtu piekļuvi Ziemassvētku eglīšu rotājumu iegādei un personalizācijai, kā arī administrācijas rīkus uzņēmuma pārvaldībai.

Mājaslapas dizains tiks veidots mūsdienīgā stilā, kas atbilst aktuālajām tīmekļa dizaina tendencēm, radot lietotnē harmonisku vidi. Tiks izvēlēta krāsu gamma un grafiskie elementi, kas uzsver svētku noskaņu un izceļ piedāvāto produkciju.

Klientu daļa būs paredzēta lietotāju ērtībai un tajā tiks iekļautas šādas funkcionalitātes:

* Jaunumi: sadaļa, kurā lietotāji varēs apskatīt aktuālo informāciju par uzņēmuma piedāvājumiem, jauniem produktiem un citiem paziņojumiem.
* Gatava produkcija: sadaļa, kurā tiks parādīti jau gatavie eglīšu rotājumi ar detalizētu aprakstu, cenām un attēliem.
* Pasūtīšana: iespēja izvēlēties un iegādāties produktus, izmantojot ērti lietojamu pasūtījuma veidlapu.
* Izejmateriālu apskate: sadaļa, kurā klienti varēs iepazīties ar materiāliem, no kuriem tiek izgatavoti rotājumi, lai veidotu personalizētus produktus.
* Savas rotaļas izveidošana: funkcionalitāte, kas ļaus klientiem personalizēt rotājumus, izvēloties krāsas, formas un dekoratīvos elementus, lai radītu unikālu dizainu.
* Saziņa ar uzņēmumu: kontakt forma, kas nodrošina ātru komunikāciju ar uzņēmuma pārstāvjiem jautājumu un ierosinājumu gadījumā.
* **Klientu atsauksmes**: sadaļa, kurā lietotāji varēs lasīt citu klientu atsauksmes par produktiem, kā arī iesniegt savas atsauksmes pēc pirkuma veikšanas. Šī funkcionalitāte palīdzēs veidot uzticību potenciālajiem klientiem un sniegs uzņēmumam iespēju uzlabot piedāvājumu, ņemot vērā klientu ierosinājumus un viedokļus.

Administrācijas daļa tiks veidota, lai nodrošinātu ērtu uzņēmuma vadību, un tajā būs šādas iespējas:

* Pasūtījumu pārvaldība: administratoriem būs iespēja apskatīt saņemtos pasūtījumus, tos rediģēt vai dzēst, ja nepieciešams, kā arī mainīt pasūtījuma statusu, piemēram, "izpildīts" vai "gaida apstiprinājumu".
* Produkcijas pārvaldība: funkcionalitāte, kas ļaus administratoriem pievienot jaunus produktus gatavo rotājumu katalogam, rediģēt esošo produkciju vai to dzēst.
* Izejmateriālu pārvaldība: iespēja pievienot jaunus materiālus, rediģēt informāciju par esošajiem vai dzēst tos no sistēmas, lai nodrošinātu aktuālu informāciju klientiem.
* Lietotāju pārvaldība: iespēja pievienot jaunus lietotājus, rediģēt informāciju par esošajiem vai dzēst tos no sistēmas.

Visas sistēmas funkcijas būs balstītas uz datu bāzi, kas nodrošinās drošu un efektīvu datu uzglabāšanu un apstrādi. Datu bāzē tiks glabāta informācija par produktiem, materiāliem, pasūtījumiem un lietotājiem, ļaujot sistēmai ātri piekļūt nepieciešamajiem datiem. Klientu un administratoru saskarnes būs savstarpēji saistītas, un izmaiņas administrācijas daļā automātiski atspoguļosies klientu daļā.

Šāds sistēmas izstrādes plāns nodrošinās ne tikai funkcionalitāti un lietotāju ērtības, bet arī efektīvu uzņēmuma darbības pārvaldību un ilgtspējīgu attīstību.

# Programmatūras prasību specifikācija

Šī sadaļa apraksta produkta perspektīvu, funkcionālās un nefunkcionālās prasības, kā arī gala lietotāja raksturiezīmes. **Produkta perspektīvā** ir vērsta uz personalizācijas iespēju paplašināšanu, videi draudzīgu materiālu izmantošanu un ilgtspējīgas attīstības principu ievērošanu, stiprinot zīmola tēlu un klientu lojalitāti. **Funkcionālās prasības** ietver vienkāršu produktu apskati, pasūtīšanu un personalizāciju, savukārt **nefunkcionālās prasības** pieprasa mājaslapas ātru lietojamību un modernu dizainu. **Gala lietotāji** ir klienti, kas novērtē unikālus un pielāgojamus produktus, un administratori, kas nodrošina sistēmas efektīvu darbību un kvalitatīvu atbalstu.

## 2.1. Produkta perspektīva

Internetveikala "Sparkly Dream" nākotnes perspektīvas ir vērstas uz produkta attīstību, klientu pieredzes pilnveidošanu un uzņēmuma ilgtspējību. Platforma var paplašināt savu piedāvājumu, ieviešot jaunus roku darba produktus, kas piemēroti dažādiem svētkiem, piemēram, Lieldienām, dzimšanas dienām vai kāzām. Tas ļaus saglabāt klientu interesi un nodrošinās pieprasījumu visa gada garumā. Turklāt mūsdienu tehnoloģiju integrācija, piemēram, papildinātās realitātes rīki, varētu piedāvāt klientiem iespēju vizualizēt personalizētos rotājumus pirms pasūtījuma veikšanas, kas uzlabos lietotāja pieredzi un veicinās uzticību.

## 2.2. Sistēmas funkcionālās prasības

### 2.2.1. Klientu reģistrācija

Mērķis: Nodrošināt iespēju klientiem izveidot kontus mājaslapā.

Ievaddati:

1.tabula

Klientu reģistrācija

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligāts | Piezīmes |
| E-pasts | Jā | Unikāls |
| Vārds | Jā |  |
| Uzvārds | Jā |  |
| Parole | Jā | Tiek saglabāts šifrētā formā |
| Foto | Nē |  |

Apstrāde: Kad reģistrācijas poga tiek nospiesta, sistēma pārbauda vai visi ievadlauki ir aizpildīti. Tālāk tiek pārbaudīti dati, vai datu bāzē neeksistē ievadītais e-pasts un lietotājvārds. Ja visi ievadlauki ir aizpildīti korekti, tad klients tiek reģistrēts, un viņa dati tiek saglabāti datu bāzē ar šifrētu paroli.

Izvaddati:

1. Kļūdas paziņojums, ja gadījumā datu bāze eksistē ievadītais e-pasts.
2. Kļūdas paziņojums, ja visi ievadlauki nav aizpildīti.
3. Paziņojums par veiksmīgu reģistrāciju.

### 2.2.2. Klientu autorizēšana sistēma

Mērķis: Funkcija nodrošina iespēju klientiem autorizēties savā kontā.

Ievaddati:

2.tabula

Klientu autorizēšana sistēmā

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligāts | Piezīmes |
| Lietotājvārds | Jā | Unikāls |
| Parole | Jā | Tiek saglabāts šifrētā formā |

Apstrāde: Kad autorizācijas poga tiek nospiesta, sistēma pārbauda vai visi ievadlauki ir aizpildīti. Tālāk tiek pārbaudīti dati, lai tie sakristu ar tiem, kas atrodas datu bāzē. Ja visi ievadlauki ir aizpildīti korekti, tad klients ir autorizēts savā kontā.

Izvaddati:

1. Kļūdas paziņojums, ja ievadītais e-pasts neeksistē datu bāze.
2. Kļūdas paziņojums, ja tiek ievadīta neparieza parole.
3. Kļūdas paziņojums, ja visi ievadlauki nav aizpildīti.

### 2.2.3. Klienta izlogošanas

Mērķis: Nodrošināt autorizētajam klientam izlogoties no sava konta

Ievaddati: Noklikšķināt uz izlogošanas pogu

Apstrāde: Sistēma pārtrauc sesiju, un atgriež klientu uz mājaslapas sākumlapu.

Izvaddati: Lietotāja atgriešana uz mājaslapas sākumlapu.

### 2.2.4. Klientu dzēšana no sistēmas

Mērķis: Nodrošināt iespēju klientam dzēst savu kontu.

Ievaddati: Klientam jābūt autorizētam, un sadaļā “Profils/Iestatījumi” jānospiež poga “Dzēst kontu”, un jāapstiprina darbība.

Apstrāde: Funkcija pārbauda vai lietotājs ir autorizēts, dzēš klientu no datu bāzes un pārtrauc sesiju, atgriežot klientu uz sākumlapu.

Izvaddati: Klients saņem paziņojumu par veiksmīgu konta dzēšanu un tiek atgriezts uz sākumlapu.

### 2.2.5. Klienta personīgas informācijas atjaunošana

Mērķis:Funkcija ir paredzēta klientiem, lai tie varētu atjaunot savu personīgo informāciju.

Ievaddati:

**3.tabula**

Dati paroles maiņai

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Vārds | Jā |  |
| Uzvārds | Jā |  |
| E-pasts | Jā |  |
| Attēls | Nē |  |

Apstrāde: Kad tiek nospiesta poga “Saglabāt”, sistēma pārbauda vai klients ar tādu lietotājvārdu eksistē un vai ir aizpildīti visi ievads lauki. Ja tā ir, tad sistēmā tiek atjauno klienta personīgo informāciju.

Izvaddati:

1) Paziņojums par veiksmīgu personīgas informācijas atjaunošanu;

2) Kļūdas ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.6. Klienta paroles maiņa

Mērķis:Funkcija ir paredzēta klientiem, lai tie varētu mainīt savu autentifikācijas paroli.

Ievaddati:

**3.tabula**

Dati paroles maiņai

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Vecā parole | Jā |  |
| Jaunā parole | Jā |  |
| Parole atkārtoti | Jā |  |

Apstrāde: Kad tiek nospiesta poga “Saglabāt”, sistēma pārbauda vai klients ar tādu lietotājvārdu eksistē un vai ievadītas paroles abos laukos sakrīt. Ja tā ir, tad sistēmā tiek pievienota jauna šifrēta parole.

Izvaddati:

1) Paziņojums par veiksmīgu paroles maiņu;

2) Kļūdas ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.7. Produkcijas apskate

Mērķis: Nodrošināt mājaslapas apmeklētajiem apskatīt esošo produkciju attiecīgajā mājaslapas sadaļā.

Apstrāde: Attiecīgajā mājaslapas sadaļā, tiek paradīta visa produkcija no datu bāzes.

Izvaddati: Kļūdas paziņojums, ja datu bāzē nav nevienas produkcijas.

### 2.2.8. Izejmateriālu apskate

Mērķis: Nodrošināt mājaslapas apmeklētajiem apskatīt izejmateriālus, no kuriem tiek veidotas rotaļlietas attiecīgajā mājaslapas sadaļā.

Apstrāde: Attiecīgajā mājaslapas sadaļā, tiek paradīti visi izejmateriāli no datu bāzes.

Izvaddati: Kļūdas paziņojums, ja datu bāzē nav neviena ieraksta.

### 2.2.9. Atsauksmju apskate

Mērķis: Nodrošināt mājaslapas apmeklētajiem apskatīt atsauksmes attiecīgajā mājaslapas sadaļā.

Apstrāde: Attiecīgajā mājaslapas sadaļā, tiek paradīta visas atsauksmes no datu bāzes.

Izvaddati: Kļūdas paziņojums, ja datu bāzē nav neviena ieraksta.

### 2.2.10. Jaunumu apskate

Mērķis: Nodrošināt mājaslapas apmeklētajiem apskatīt jaunumus attiecīgajā mājaslapas sadaļā.

Apstrāde: Attiecīgajā mājaslapas sadaļā, tiek paradīti visi jaunumi no datu bāzes.

Izvaddati: Kļūdas paziņojums, ja datubāzē nav neviena ieraksta.

### 2.2.11. Produktu pievienošana iepirkuma grozā

Mērķis: Nodrošināt klientiem pievienot izvēlēto produktu savā iepirkuma grozā.

Ievaddati: Pie izvēlēta produkta klikšķināt uz pogu “Pievienot grozam”.

Apstrāde: Sistēma pārbauda, vai klients ir autorizēts, ja tā ir, tad produkts tiek pievienots iepirkuma grozā, pretēja gadījuma, sistēma atver ielogošanas lapu.

Izvaddati:

1. Kļūdas paziņojums, ja klients nav autorizēts.

2) Paziņojums par veiksmīgu pievienošanu grozā.

### 2.2.12. Produktu dzēšana no iepirkuma grozā

Mērķis: Nodrošināt iespēju klientiem dzēst produktu no iepirkuma groza.

Ievaddati: Klients klikšķina uz dzēšanas pogu pie konkrēta produkta, ko vēlās dzēst no iepirkuma groza.

Apstrāde: Produkts tiek izdzēsts no klienta iepirkuma groza.

Izvaddati: Paziņojums par veiksmīgu produkta dzēšanu no iepirkuma groza.

### 2.2.13. Produktu daudzuma mainīšana iepirkuma grozā

Mērķis: Nodrošināt iespēju klientiem nomainīt konkrēta produkta daudzumu iepirkuma grozā.

Ievaddati: Klients klikšķina uz “+” vai “-” pie konkrēta produkta, kuram vēlās nomainīt daudzumu.

Apstrāde: Produkta daudzums tiek nomainīts.

Izvaddati: Paziņojums par veiksmīgu produkta daudzuma nomainīšanu iepirkuma grozā.

### 2.2.14. Visa groza iztīrīšana

Mērķis: Nodrošināt iespēju klientiem iztīrīt visu grozu vienā piegājienā.

Ievaddati: Klients klikšķina uz “Iztīrīr grozu” pogu.

Apstrāde: Kienta iepirkuma grozs tiek iztīrīts.

Izvaddati: Paziņojums par veiksmīgu produkta dzēšanu no iepirkuma groza.

### 2.2.15. Pasūtījuma veikšana

Mērķis: Nodrošināt klientiem veikt pasutījumu.

Ievaddati:

6.tabula

Pasūtījuma veikšana

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| E-pasts | Jā |  |
| Vārds | Jā |  |
| Uzvārds | Jā |  |
| Tālrunis | Jā |  |
| Adrese | Jā |  |
| Kartes numurs | Jā |  |
| Kartes derīguma datums | Jā |  |
| Kartes CVC kods | Jā |  |

Apstrāde: Sistēma pārbauda, vai visi ievadlauki ir aizpildīti, un pievieno pasūtījumu datu bāzē ar statusu “Iesniegts”.

Izvaddati:

1. Kļūdas paziņojums, ja visi ievadlauki nav aizpildīti.
2. Paziņojums, par veiksmīgu pasūtījumu.

### 2.2.16. Savas rotaļas izveide

Mērķis: Sadaļā “Izveido pats” nodrošināt iespēju klientam izvēlēties izejmateriālus, lai izveidotu savu rotaļu.

Ievaddati:

**5.tabula**

**Savas rotaļas izveide**

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Forma | Jā | Izvēlēties no saraksta |
| Pamatkrāsa | Jā | Izvēlēties no saraksta |
| Mālu figūra | Nē | Izvēlēties no saraksta |
| Dekorējums | Jā | Izvēlēties no saraksta |
| Vārds | Jā |  |
| Uzvārds | Jā |  |
| E-pasts | Jā |  |
| Tālrunis | Jā |  |
| Adrese | Jā |  |
| Pilsēta | Jā |  |
| Pasta indekss | Jā |  |
| Piegāde | Jā |  |
| Apmaksas veids | Jā |  |

Apstrāde: Sistēma pārbauda vai klients ir autorizēts, ja tā ir, tad saglabā klienta izvēlētus parametrus datu bāzē un parāda tos iepirkuma grozā. Ja klients nav autorizēts, tad sistēma atver ielogošanas logu.

Izvaddati: Paziņojums par veiksmīgu rotaļas izveidi.

### 2.2.17. Saziņa ar pārdevēju

Mērķis: Nodrošināt iespēju klientam sazināties ar pārdevēju caur e-pastu.

Ievaddati:

**4.tabula**

**Saziņa ar pārdevēju**

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Vārds, Uzvārds | Jā |  |
| E-pasts | Jā |  |
| Tālrunis | Nē |  |
| Ziņojuma teksts | Jā |  |

Apstrāde: Kad ir nospiesta poga “Sazināties”, sistēma pārbauda visus obligāti aizpildāmos laukus un datu tipus. Ja tā ir, ziņa tiek nosūtīta uz organizācijas e-pastu.

Izvaddati:

1) Kļūdas paziņojums, ja nav aizpildīti visi ievades lauki.

2) Paziņojums par veiksmīgu e-pasta nosūtīšanu.

### 2.2.18. Administratoru un moderatoru ielogošanas sistēma

Mērķis: Funkcija nodrošina iespēju administratoriem un moderatoriem ielogoties savā kontā.

Ievaddati:

7.tabula

Administratoru un moderatoru ielogošanas sistēma

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligāts | Piezīmes |
| Lietotājvārds | Jā | Unikāls |
| Parole | Jā | Tiek saglabāts šifrētā formā |

Apstrāde: Kad autorizācijas poga tiek nospiesta, sistēma pārbauda vai visi ievadlauki ir aizpildīti. Tālāk tiek pārbaudīti dati, lai tie sakristu ar tiem, kas atrodas datu bāzē. Ja visi ievadlauki ir aizpildīti korekti, tad lietotājam atvērās atbilstoši viņa lomai sadaļas. .

Izvaddati:

1. Kļūdas paziņojums, ja ievadītais lietotājvārds vai parole nesakrīt ar datu bāzi.
2. Kļūdas paziņojums, ja visi ievadlauki nav aizpildīti.

### 2.2.19. Lietotāja pievienošana sistēmā

Mērķis: Funkcija paredzēta autorizētajam administratoram pievienot jaunu lietotāju sistēmā.

Ievaddati:

8.tabula

Lietotāja pievienošana sistēmā

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| E-pasts | Jā | Unikāls |
| Vārds | Jā |  |
| Uzvārds | Jā |  |
| Lietotājvārds | Jā |  |
| Tālrunis | Jā |  |
| Parole | Jā | Tiek saglabāts šifrētā formā |
| Loma | Jā | Izvēlēties no saraksta |
| Foto | Nē |  |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma pārbauda, ka visi ievadlauki ir aizpildīti, ka arī pārbauda, vai datu bāzē neeksistē ievadītais e-pasts un lietotājvārds. Ja viss ir aizpildīts korekti, tad jaunais lietotājs tiek saglabāts datu bāzē.

Izvaddati:

1. Kļūdas paziņojums, ja visi ievadlauki nav aizpildīti.
2. Kļūdas paziņojums, ja datu bāzē jau eksistē ievadītais e-pasts vai lietotājvārds.
3. Paziņojums par veiksmīgu lietotāja pievienošanu.

### 2.2.20. Lietotāja izlogošanas

Mērķis: Nodrošināt autorizētajam lietotājam izlogoties no sava konta

Ievaddati: Noklikšķināt uz izlogošanas pogu

Apstrāde: Sistēma pārtrauc sessiju, un atgriež lietotāju uz autorizēšanas logu.

Izvaddati: Lietotāja atgriešana uz autorizēšanas logu.

### 2.2.21. Lietotāja dzēšana no sistēmas

Mērķis: Funkcija paredzēta autorizētajam administratoram dzēst lietotāju no sistēmas.

Ievaddati: Pie konkrēta lietotāja jānospiež poga “Dzēst”.

Apstrāde: Sistēma prasa apstiprinājumu, un pēc tā dzēš lietotāju.

Izvaddati: Paziņojums par veiksmīgu lietotāja dzēšanu.

### 2.2.22. Produkta pievienošana

Mērķis: Nodrošināt sistēmas administratoriem vai moderatoriem iespēju pievienot jaunus produktus.

Ievaddati:

**9.tabula**

**Produkta pievienošana**

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Nosaukums | Jā |  |
| Forma | Jā | Izvēlēties no saraksta |
| Pamatkrāsa | Jā | Izvēlēties no saraksta |
| Dekorējums1 | Jā | Izvēlēties no saraksta |
| Dekorejums2 | Jā | Izvēlēties no saraksta |
| Mālu figūra | Jā | Izvēlēties no saraksta |
| Apraksts | Jā |  |
| Attēls1 | Jā |  |
| Attēls2 | Jā |  |
| Attēls3 | Jā |  |
| Cena | Jā |  |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu produkta pievienošanu sistēmai.

2) Kļūdas paziņojums, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.23. Produkta rediģēšana

Mērķis: Nodrošināt sistēmas lietotājiem rediģēt produktu.

Ievaddati: Skat. 8. tabulu

Apstrāde: Kad poga “Saglabāt” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu produkta rediģēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.24. Produktu dzēšana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju dzēst produktus.

Ievaddati: Lietotājs klikšķina uz pogu “Dzēst” pie konkrēta produkta, ko vēlās dzēst.

Apstrāde: Kad poga “Dzēst” tiek nospiesta, sistēma dzēš visu ierakstu par noteiktu produktu.

Izvaddati:

1) Paziņojums par veiksmīgu produkta dzēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.25. Jaunumu pievienošana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju pievienot jaunumus.

Ievaddati:

**10.tabula**

**Jaunumu pievienošana**

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Nosaukums | Jā |  |
| Teksts | Jā |  |
| Attēls | Jā |  |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu jaunuma pievienošanu sistēmai.

2) Kļūdas paziņojums, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.26. Jaunumu rediģēšana

Mērķis: Nodrošināt sistēmas lietotājiem rediģēt jaunumus.

Ievaddati: Skat. 9. tabulu

Apstrāde: Kad poga “Saglabāt” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu jaunuma rediģēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.27. Jaunumu dzēšana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju dzēst jaunumus.

Ievaddati: Lietotājs klikšķina uz pogu “Dzēst” pie konkrēta jaunuma, ko vēlās dzēst.

Apstrāde: Kad poga “Dzēst” tiek nospiesta, sistēma dzēš visu ierakstu par noteiktu jaunumu.

Izvaddati:

1) Paziņojums par veiksmīgu jaunumu dzēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.28. Rotaļlietas formas pievienošana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju pievienot rotaļlietas formas.

Ievaddati:

11.tabula

Rotaļlietas formas pievienošana

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Forma | Jā |  |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu formas pievienošanu sistēmai.

2) Kļūdas paziņojums, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.29. Rotaļlietas formas rediģēšana

Mērķis: Nodrošināt sistēmas lietotājiem rediģēt rotaļlietas formas.

Ievaddati: Skat. 11. tabulu

Apstrāde: Kad poga “Saglabāt” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu rotaļlietas formas rediģēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.30. Rotaļlietas formas dzēšana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju dzēst rotaļlietas formas.

Ievaddati: Lietotājs klikšķina uz pogu “Dzēst” pie konkrēta rotaļlietas formas, ko vēlās dzēst.

Apstrāde: Kad poga “Dzēst” tiek nospiesta, sistēma dzēš visu ierakstu par noteiktu rotaļlietas formu.

Izvaddati:

1) Paziņojums par veiksmīgu rotaļlietas formas dzēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.31. Rotaļlietas auduma pievienošana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju pievienot rotaļlietas audumu.

Ievaddati:

12.tabula

Rotaļlietas auduma pievienošana

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Audums | Jā |  |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu auduma pievienošanu sistēmai.

2) Kļūdas paziņojums, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.32. Rotaļlietas auduma rediģēšana

Mērķis: Nodrošināt sistēmas lietotājiem rediģēt rotaļlietas audumu.

Ievaddati: Skat. 11. tabulu

Apstrāde: Kad poga “Saglabāt” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu auduma rediģēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.33. Rotaļlietas auduma dzēšana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju dzēst rotaļlietas audumu.

Ievaddati: Lietotājs klikšķina uz pogu “Dzēst” pie konkrēta rotaļlietas formas, ko vēlās dzēst.

Apstrāde: Kad poga “Dzēst” tiek nospiesta, sistēma dzēš visu ierakstu par noteiktu audumu.

Izvaddati:

1) Paziņojums par veiksmīgu rotaļlietas auduma dzēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.34. Rotaļlietas dekorējuma (1) pievienošana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju pievienot rotaļlietas dekorējumu (1).

Ievaddati:

13.tabula

Rotaļlietas dekorējuma (1).pievienošana

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Dekorējums (1) | Jā |  |
| Attēls | Jā |  |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu dekorējuma (1).pievienošanu sistēmai.

2) Kļūdas paziņojums, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.35. Rotaļlietas dekorējuma (1).rediģēšana

Mērķis: Nodrošināt sistēmas lietotājiem rediģēt rotaļlietas dekorējumu (1).

Ievaddati: Skat. 11. tabulu

Apstrāde: Kad poga “Saglabāt” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu dekorējuma (1).rediģēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.36. Rotaļlietas dekorējuma (1).dzēšana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju dzēst rotaļlietas dekorējumu (1).

Ievaddati: Lietotājs klikšķina uz pogu “Dzēst” pie konkrēta rotaļlietas formas, ko vēlās dzēst.

Apstrāde: Kad poga “Dzēst” tiek nospiesta, sistēma dzēš visu ierakstu par noteiktu dekorējumu (1).

Izvaddati:

1) Paziņojums par veiksmīgu rotaļlietas dekorējuma (1) dzēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.37. Rotaļlietas dekorējuma (2) pievienošana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju pievienot rotaļlietas dekorējumu (1).

Ievaddati:

14.tabula

Rotaļlietas dekorējuma (2).pievienošana

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| Dekorējums (2) | Jā |  |
| Attēls | Jā |  |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu dekorējuma (2).pievienošanu sistēmai.

2) Kļūdas paziņojums, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.38. Rotaļlietas dekorējuma (2).rediģēšana

Mērķis: Nodrošināt sistēmas lietotājiem rediģēt rotaļlietas dekorējumu (2).

Ievaddati: Skat. 11. tabulu

Apstrāde: Kad poga “Saglabāt” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu dekorējuma (2).rediģēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.39. Rotaļlietas dekorējuma (2).dzēšana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju dzēst rotaļlietas dekorējumu (2).

Ievaddati: Lietotājs klikšķina uz pogu “Dzēst” pie konkrēta rotaļlietas formas, ko vēlās dzēst.

Apstrāde: Kad poga “Dzēst” tiek nospiesta, sistēma dzēš visu ierakstu par noteiktu dekorējumu (2).

Izvaddati:

1) Paziņojums par veiksmīgu rotaļlietas dekorējuma (2) dzēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.40. Klientu pievienošana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju pievienot klientus.

Ievaddati:

**15.tabula**

**Klientu pievienošana**

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| E-pasts | Jā | Unikāls |
| Vārds | Jā |  |
| Uzvārds | Jā |  |
| Tālrunis | Jā |  |
| Parole | Jā | Tiek saglabāts šifrētā formā |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu klienta pievienošanu sistēmai.

2) Kļūdas paziņojums, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.41. Klienta rediģēšana

Mērķis: Nodrošināt sistēmas lietotājiem rediģēt klientus.

Ievaddati: Skat. 12. tabulu

Apstrāde: Kad poga “Saglabāt” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu klienta rediģēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.42. Klientu dzēšana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju dzēst klientus.

Ievaddati: Lietotājs klikšķina uz pogu “Dzēst” pie konkrēta klienta, ko vēlās dzēst.

Apstrāde: Kad poga “Dzēst” tiek nospiesta, sistēma dzēš visu ierakstu par noteiktu klientu.

Izvaddati:

1) Paziņojums par veiksmīgu klienta dzēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.43. Pasūtījuma pievienošana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju pievienot pasūtījumus.

Ievaddati:

16.tabula

Pasūtījuma pievienošana

|  |  |  |
| --- | --- | --- |
| **Nosaukums** | **Obligāts** | **Piezīmes** |
| E-pasts | Jā |  |
| Vārds | Jā |  |
| Uzvārds | Jā |  |
| Tālrunis | Jā |  |
| Adrese | Jā |  |

Apstrāde: Kad poga “Pievienot” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu pasūtījuma pievienošanu sistēmai.

2) Kļūdas paziņojums, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.44. Pasūtījuma rediģēšana

Mērķis: Nodrošināt sistēmas lietotājiem rediģēt pasūtījumus.

Ievaddati: Skat. 13. tabulu

Apstrāde: Kad poga “Saglabāt” tiek nospiesta, sistēma veic pārbaudi, vai visi obligāti lauki ir aizpildīti un pārbauda visus datu tipus.

Izvaddati:

1) Paziņojums par veiksmīgu pasūtījuma rediģēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

### 2.2.45. Pasūtījuma dzēšana

Mērķis: Nodrošināt sistēmas lietotājiem iespēju dzēst klientus.

Ievaddati: Lietotājs klikšķina uz pogu “Dzēst” pie konkrēta pasūtījuma, ko vēlās dzēst.

Apstrāde: Kad poga “Dzēst” tiek nospiesta, sistēma dzēš visu ierakstu par noteiktu klientu.

Izvaddati:

1) Paziņojums par veiksmīgu klienta dzēšanu sistēmā.

2) Kļūdu ziņojumi, ja procesā radusies kļūda vai nepieciešamas papildu darbības.

## 2.3. Sistēmas nefunkcionālās prasības

1) Tīmekļa vietne ir izstrādāta latviešu valodā, lai nodrošinātu tās piemērotību vietējiem lietotājiem un viegli saprotamu informāciju dažādām sabiedrības grupām neatkarīgi no lietotāju vecuma, izglītības vai tehniskajām zināšanām.

2) Mājaslapas dizains ir estētisks, mūsdienīgs un viegli pārskatāms, lai mājaslapas apmeklētajiem būtu viegli mijiedarboties.

3) Mājaslapa ir responsīva visām ierīcēm.

4) Mājaslapa ir pilnībā funkcionāla visās populārākajās pārlūkprogrammās (Chrome, Firefox, Edge, Safari).

5) Katra mājaslapas sadaļa tiek atvērta jaunajā logā.

## 2.4. Gala lietotāja raksturiezīmes

Mājaslapas lietotājiem nav īpaši jābūt datora prasmēm, tāpēc ka mājaslapa ir uztaisīta ar vieglu un saprotamu interfeisu, gan lietotājiem, gan administratoriem.

* Klienti: Šajā mājaslapa klienta tiesības ir apskatīt visu publisko mājaslapu, veikt pasūtījumus, izstrādāt savu īpašu rotaļu un sazināties ar organizāciju.
* Administratori: Administratora tiesības ir CRUD (Create, read, update, edit) operācijas saistībā ar klientiem, lietotājiem, produkciju, izejmateriāliem, pasūtījumiem un jaunumiem.
* Moderatori: Moderatora tiesības ir CRUD (Create, read, update, edit) operācijas saistībā ar klientiem, produkciju, izejmateriāliem, pasūtījumiem un jaunumiem.

# Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

Lai nodrošinātu "Sparkly Dream" platformas augstu veiktspēju, drošību un lietošanas ērtumu, ir rūpīgi izvēlēti piemēroti izstrādes rīki un tehnoloģijas. Šajā sadaļā tiek apskatīti izmantotie programmēšanas rīki, datubāzes risinājumi, dizaina un lietotāja interfeisa izstrādes instrumenti, kā arī izvēles pamatojums, kas balstīts uz projekta prasībām un tehniskajām iespējām.

## 3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

* **HMTL**

HTML (HyperText Markup Language) ir izmantots kā galvenais valodas pamats "Sparkly Dream" platformas lietotāja saskarnes izveidei. Tas nodrošina mājaslapas struktūru un ļauj efektīvi organizēt satura izvietojumu, kas ir būtiski ērtai navigācijai un lietotāju pieredzei.

* **CSS**

CSS (Cascading Style Sheets) ir izmantots, lai nodrošinātu "Sparkly Dream" mājaslapas vizuālo noformējumu un lietotāja pieredzi. Ar CSS palīdzību tiek definēti krāsu salikumi, fonti, izkārtojums un animācijas, kas padara mājaslapu pievilcīgu un viegli lietojamu. Responsīvais dizains nodrošina pareizu mājaslapas attēlošanu dažādās ierīcēs, tostarp datoros, planšetēs un viedtālruņos.

* **JavaScript**

JavaScript ir populāra programmēšanas valoda, kas tiek izmantota tīmekļa izstrādē, lai pievienotu interaktivitāti un dinamisku funkcionalitāti mājaslapām. Tā darbojas lietotāja pārlūkprogrammā un ļauj manipulēt ar HTML un CSS elementiem reāllaikā. JavaScript nodrošina iespēju veidot dinamiskas saskarnes, uzlabot lietotāja pieredzi un efektīvi apstrādāt datus bez nepieciešamības pārlādēt lapu.

JavaScript tiek izmantots "Sparkly Dream" mājaslapā šādiem mērķiem:

1) Animācijas un lietotāja mijiedarbība:

JavaScript tiks izmantots, lai pievienotu dažādas animācijas, piemēram, vieglus pārejas efektus, modālos logus un hover efektus, izmantojot bibliotēkas, piemēram, GSAP vai tikai CSS klasēm pievienotas un noņemtas klases ar JavaScript.

2) Asinhrona datu ielādē (AJAX, Fetch API):

Lai uzlabotu lapas ielādēs ātrumu un lietotāja pieredzi, tiks izmantots JavaScript, lai asinhroni iegūtu un atjauninātu datus no servera, nenoslogojot visu lapu.

* **PHPMyAdmin**

phpMyAdmin ir tīmekļa lietojumprogramma, kas nodrošina MySQL datubāzes pārvaldību, izmantojot grafisko saskarni. "Sparkly Dream" projektā phpMyAdmin tiek izmantots, lai:

1) Pārvaldītu datubāzi, kas satur informāciju par lietotājiem, pasūtījumiem, produktiem un citiem sistēmas datiem.

2) Izpildītu SQL vaicājumus, lai pievienotu, rediģētu vai dzēstu ierakstus datubāzē.

phpMyAdmin ir izvēlēts, jo tas ir viegli lietojams, sniedz plašas funkcionalitātes iespējas un atbalsta MySQL/MariaDB datubāzes, kas tiek izmantotas "Sparkly Dream" projekta ietvaros.

* **PHP**

PHP (Hypertext Preprocessor) ir servera puses programmēšanas valoda, kas tiek izmantota "Sparkly Dream" platformā, lai nodrošinātu dinamisku un drošu datu apstrādi. Galvenās PHP funkcijas projektā ietver:

1) Datu apstrāde un loģika:

PHP tiek izmantots, lai apstrādātu lietotāju pieprasījumus, veiktu pasūtījumu pārvaldību un īstenotu autentifikācijas sistēmu.

2) Savienojums ar MySQL datubāzi:

PHP darbojas kā starpnieks starp mājaslapas saskarni un MySQL datubāzi, ļaujot veikt CRUD (Create, Read, Update, Delete) operācijas ar klientu datiem, pasūtījumiem un produktiem.

3) Formu validācija un drošība:

PHP tiek izmantots, lai validētu lietotāju ievadītos datus, novērstu SQL injekcijas un nodrošinātu drošu lietotāju pieredzi.

* **SQL**

SQL (Structured Query Language) ir valoda, kas tiek izmantota datu bāzu pārvaldībai un manipulācijai. "Sparkly Dream" projektā SQL tiek izmantots, lai veiktu svarīgākās datu bāzes operācijas, piemēram, datu pievienošanu, lasīšanu, atjaunināšanu un dzēšanu (CRUD operācijas). Tas nodrošina efektīvu un strukturētu piekļuvi datiem, kas nepieciešami lietotāju kontu pārvaldībai, pasūtījumu izsekošanai, produktu pievienošanai un citām sistēmas funkcijām.

* **Visual Studio Code**

Visual Studio Code ir populārs un jaudīgs koda redaktors, kas tiek plaši izmantots programmēšanas un tīmekļa izstrādes projektos. Tas piedāvā plašu funkcionalitāti, piemēram, koda krāsu atzīmēšanu, automātisko pabeigšanu, kļūdu atklāšanu un atbalstu daudziem programmēšanas valodām. Ar pieejamajiem paplašinājumiem, piemēram, atkļūdošanas rīkiem, versiju kontroles atbalstu un integrāciju ar citiem izstrādes rīkiem, Visual Studio Code padara koda rakstīšanu un pārvaldību vienkāršāku un efektīvāku. Pateicoties tā vieglajam izmēram un ātrai veiktspējai, tas ir piemērots gan maziem, gan lieliem projektiem, nodrošinot labu līdzsvaru starp funkcionalitāti un lietotāja pieredzi.

* **Github**

"Sparkly Dream" projektā GitHub tiek izmantots kā versiju kontroles rīks, lai nodrošinātu projekta koda pārvaldību un saglabātu tā organizētību. Ar GitHub palīdzību tiek sekots visām izmaiņām kodā, ļaujot viegli pārvaldīt dažādas koda versijas un atgriezties pie iepriekšējiem darba posmiem, ja nepieciešams. GitHub arī ļauj izstrādātājam vienkārši veikt izmaiņas un saglabāt tās drošā veidā, nodrošinot projektu ar versiju vēsturi un iespēju apstrādāt izmaiņas bez riska tās pazaudēt.

* **Stripe API**

Stripe API ir populārs maksājumu apstrādes rīks, kas ļauj tiešsaistes platformām un lietotnēm droši apstrādāt maksājumus. Stripe piedāvā plašu funkcionalitāti, kas ietver maksājumu pieņemšanu ar kredītkartēm, debetkartēm, elektroniskajiem maciņiem (piemēram, Apple Pay, Google Pay) un pat dažādām banku pārskaitījumu metodēm. Tā ir elastīga un viegli integrējama platforma, kas nodrošina vienkāršu un drošu risinājumu uzņēmumiem, kas vēlas veikt maksājumus tiešsaistē. "Sparkly Dream" projektā Stripe API tiks izmantots, lai apstrādātu maksājumus par pasūtījumiem, kas tiek veikti platformā.

* **PHP Mailer**

PHP Mailer ir plaši izmantota bibliotēka, kas ļauj nosūtīt e-pastus izmantojot PHP, tā vienkāršo e-pasta funkcionalitātes integrēšanu. "Sparkly Dream" projektā PHP Mailer tiks izmantots, lai nodrošinātu komunikāciju starp platformu un klientiem ļaujot lietotājiem viegli nosūtīt ziņojumus vai paziņojumus organizācijai.

## 3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts

* **Markdown**

Markdown ir vienkāršs un viegls marķējuma valoda, kas tiek izmantota, lai rakstītu tekstu tīmekļa lapām, bieži tiek izmantots dokumentācijas un saturu rakstīšanai. Tas ļauj ātri formatēt tekstu, piemēram, pievienot virsrakstus, sarakstus un linkus.

Kāpēc netiks izmantots: Markdown nav piemērots sarežģītākas struktūras vai dinamisku tīmekļa lapu veidošanai, jo tas ir ierobežots tikai tekstu un vienkāršu formātu veidošanā, tāpēc tas nav ideāli piemērots "Sparkly Dream" projektam, kur nepieciešams plašs un dinamisks HTML atbalsts.

* **Python**

Python ir augstākā līmeņa, interpretēta programmēšanas valoda, kas ir ļoti populāra tīmekļa izstrādē, datu apstrādē, mākslīgā intelekta un automatizācijas risinājumos. Tas piedāvā plašu bibliotēku un ietvaru izvēli, piemēram, Django, Flask, un FastAPI, kas ļauj viegli veidot tīmekļa lietotnes, apstrādāt datus un izveidot REST API.

Kāpēc netiks izmantots: Lai gan Python ir ļoti elastīgs un piemērots daudziem lietojumiem, "Sparkly Dream" projektā varētu būt labāk izmantot PHP, kas ir jau iekļauts tehnoloģiju kaudzē, un tas ir īpaši piemērots tīmekļa serveru un dinamisku tīmekļa lapu izveidei. Python ir nedaudz smagāks un var radīt liekus sarežģījumus, īpaši ja Python nav galvenā valoda tīmekļa servera izstrādē jūsu izvēlētajā vidē. Python varētu būt piemērots citas funkcionalitātes pievienošanai, bet šajā gadījumā PHP ir vienkāršāks un efektīvāks risinājums.

* **Tailwind CSS**

Tailwind CSS ir utilitāšu pamatots CSS ietvars, kas ļauj ļoti ātri izveidot tīmekļa lapas dizainu, izmantojot nelielas CSS klases tieši HTML elementiem.

Kāpēc netiks izmantots: Tailwind CSS var būt grūti pielāgojams, ja nepieciešama sarežģītāka dizaina izstrāde, jo tā pieeja ir ļoti atkarīga no tiešiem stilistikas iestatījumiem HTML, kas var apgrūtināt lielāku pielāgojumu un atbalsta veidošanu.

* **Node.js**

Node.js ļauj JavaScript darboties servera pusē, piedāvājot ļoti ātru un mērogojamu risinājumu tīmekļa lietotnēm.

Kāpēc netiks izmantots: Node.js var būt ļoti piemērots ļoti mērogojamām aplikācijām, bet PHP jau labi atbilst "Sparkly Dream" vajadzībām, piedāvājot ļoti stabilu un pārbaudītu risinājumu servera puses datu apstrādei, kas ir vieglāk uzturams un vairāk atbalstīts izvēlētajā tehnoloģiju kaudzē.

* **JetBrains WebStorm**

WebStorm ir pilnīga integrētā attīstības vide (IDE), kas piedāvā plašas iespējas JavaScript un Node.js izstrādei.

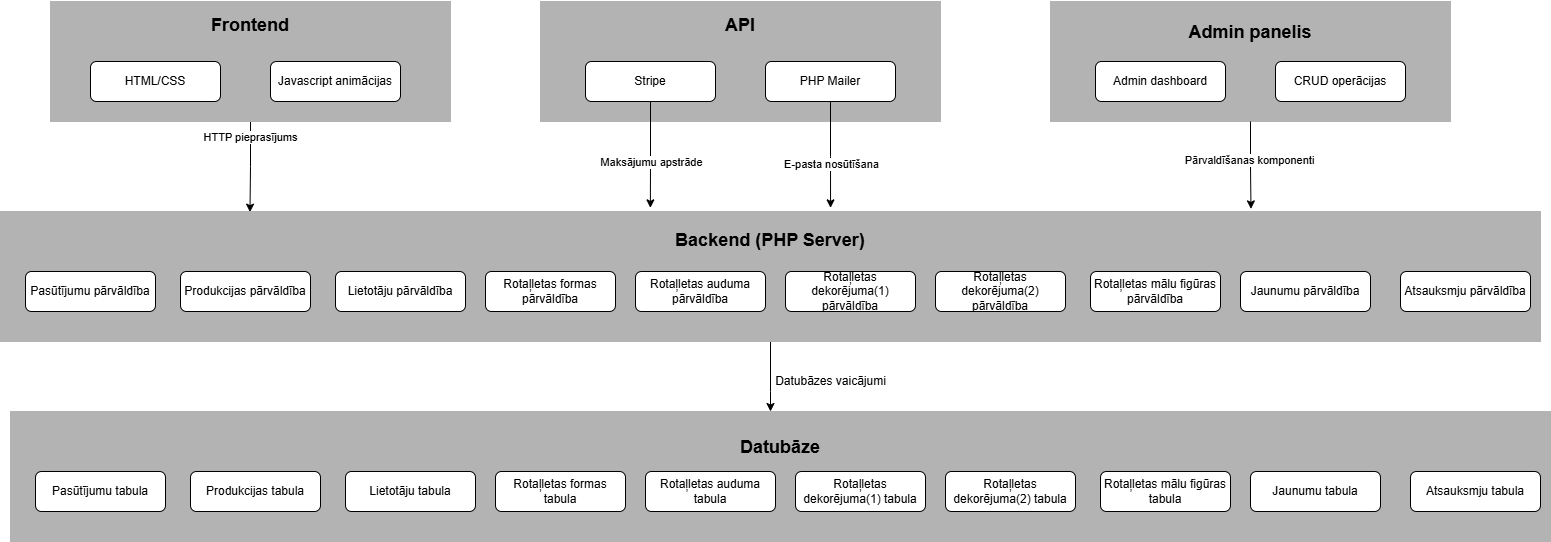
Kāpēc netiks izmantots: WebStorm var būt pārāk smags un lēns salīdzinājumā ar vieglāku, ātrāku un elastīgāku Visual Studio Code, kas ļauj ātri veikt izmaiņas un ir ļoti pielāgojams.

# Sistēmas modelēšana un projektēšana

## 4.1. Sistēmas struktūras modelis

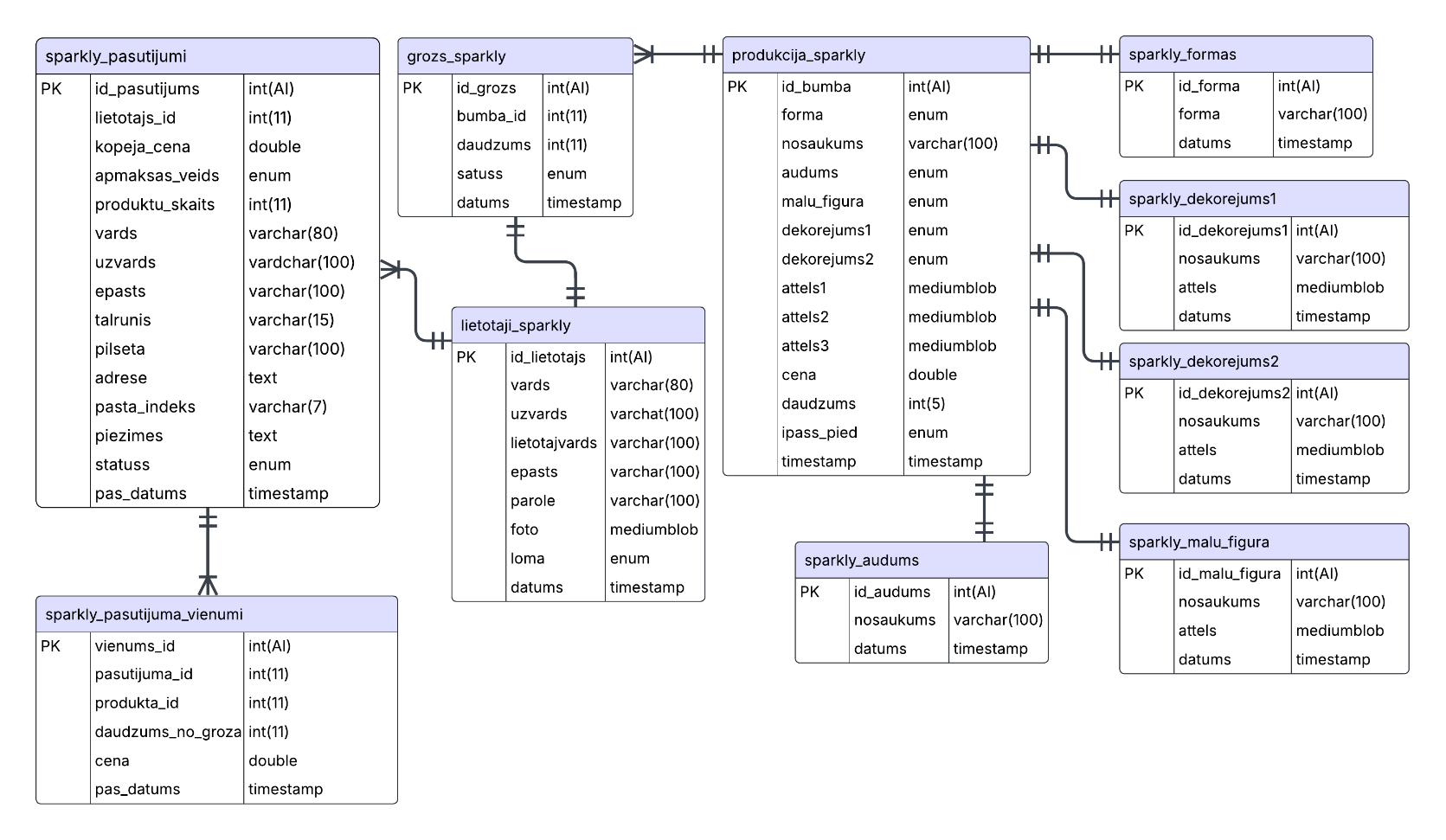
### 4.1.1. Izvietojuma (Deployment) diagramma

Izvietojuma (deployment) diagramma attēlo "Sparkly Dream" sistēmas arhitektūru. Tajā redzami četri galvenie slāņi: Frontend (HTML/CSS, JavaScript), API (Stripe, PHP Mailer), Backend (PHP serveris ar dažādām pārvaldības funkcijām) un Datubāze ar attiecīgajām tabulām. Diagramma parāda, kā komponentes mijiedarbojas un kā notiek datu plūsma starp tām, kas ir būtiski sistēmas arhitektūras izpratnei un izstrādei.

1.attēls. Izvietojuma diagramma

### 4.1.2. ER (Entity Relationship) diagramma

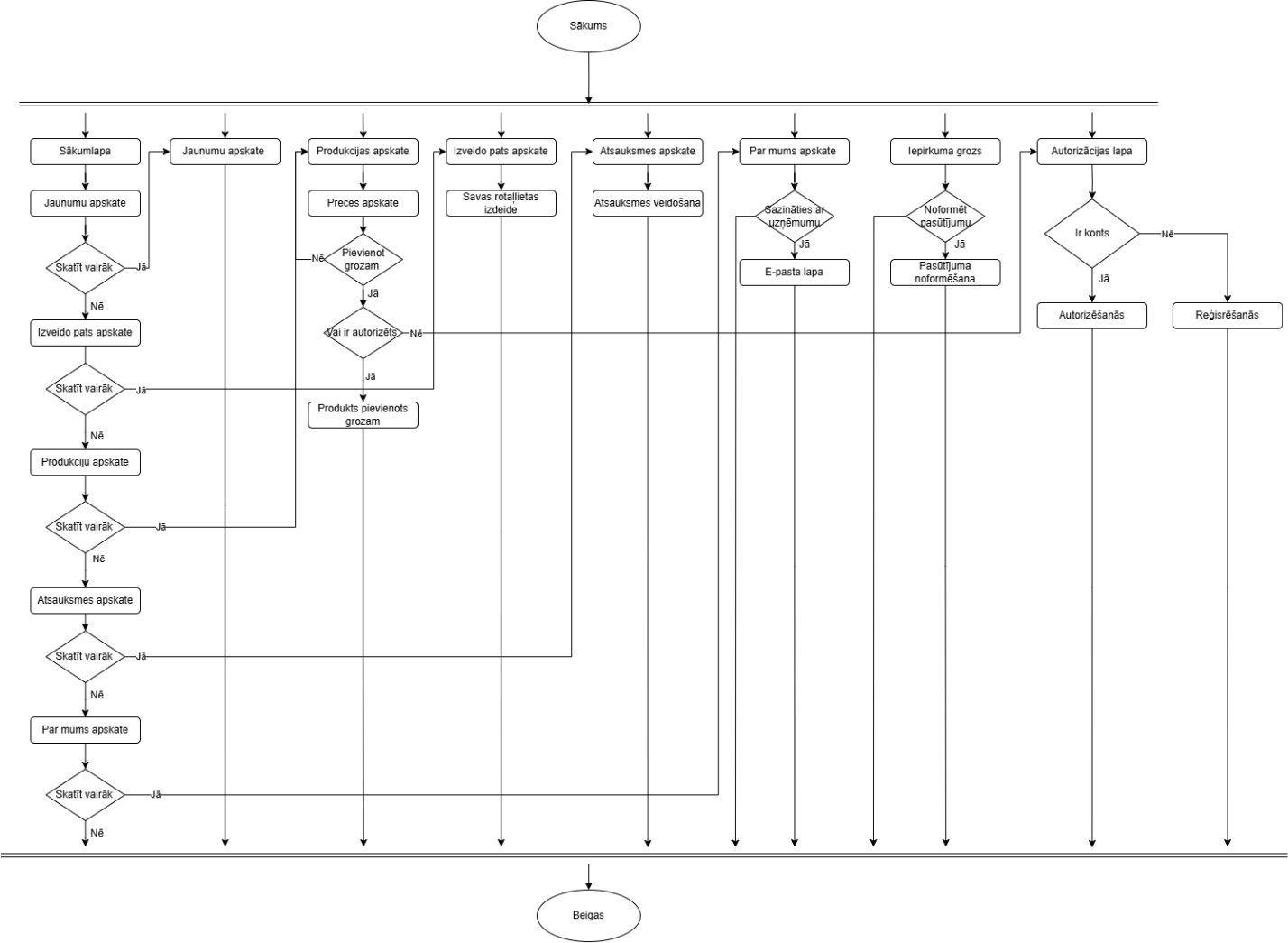
ER (Entity Relationship) diagramma attēlo "Sparkly Dream" internetveikala datubāzes struktūru. Tajā redzamas deviņas savstarpēji saistītas tabulas, kas veido sistēmas datu glabāšanas pamatu. Diagramma parāda katras tabulas laukus, datu tipus un primārās atslēgas (PK), kā arī tabulu savstarpējās attiecības, kas attēlotas ar līnijām un savienojumiem starp tabulām. Šāda struktūra nodrošina efektīvu datu organizāciju, lai atbalstītu internetveikala funkcionalitāti, ieskaitot lietotāju pārvaldību, produktu pārvaldību, pasūtījumu apstrādi un pielāgojamo rotājumu komponentu uzglabāšanu.



2.attēls. ER diagramma

## 4.2. Funkcionālais un dinamiskais sistēmas modelis

### 4.2.1. Aktivitāšu (Activity) diagramma

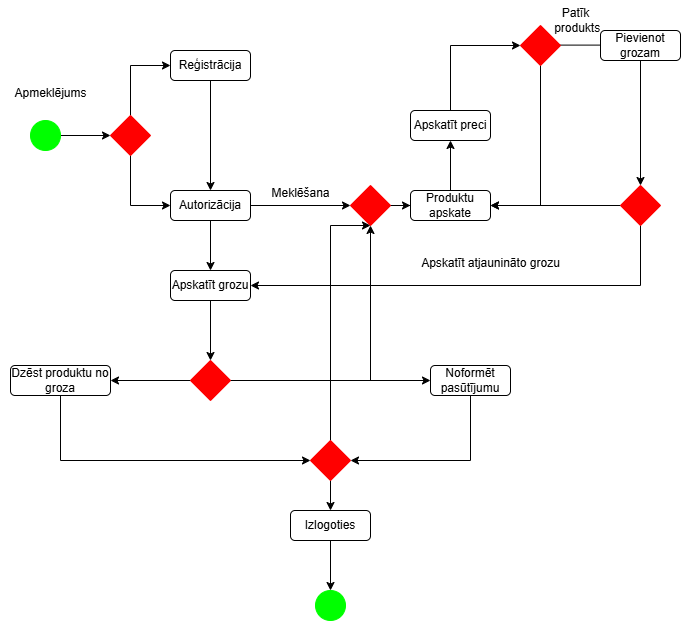
Aktivitāšu (activity) diagramma, kas pieder pie UML (Unified Modeling Language) modeļiem. Tā attēlo lietotāja ceļu caur "Sparkly Dream" internetveikala sistēmu, parādot secību, kādā notiek darbības un lēmumu pieņemšana. Šādu diagrammu izmanto, lai skaidri vizualizētu biznesa procesus un lietotāja mijiedarbību ar sistēmu. Tā ir noderīga programmatūras izstrādes procesā, lai labāk izprastu lietotāja pieredzi un identificētu sistēmas prasības, kā arī palīdz dokumentēt sistēmas darbības plūsmu tehniskajā dokumentācijā.

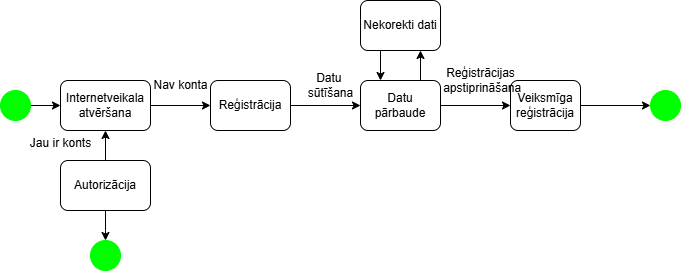
### 4.2.2. Lietojumgadījumu (Use Case) diagramma

Lietojumgadījumu diagramma attēlo "Sparkly Dream" internetveikala lietotāju lomas (viesis, lietotājs, moderators, administrators) un katrai pieeejamās funkcijas. Krāsas norāda dažādās lietotāju grupas, ar līnijām parādot pieejamās darbības katrai lomai. Diagramma palīdz izstrādātājiem organizēt sistēmas funkcionalitāti un piekļuves tiesības.

4.attēls. Lietojumgadījumu diagramma

### 4.2.3. Stāvokļu (State) diagramma (Produkta pievienōšana grozā)

Šīs ir stāvokļu (State) diagrammas, kas attēlo lietotāja iespējamās darbības un pārejas "Sparkly Dream" internetveikalā. Diagramma vizualizē klienta sesijas plūsmu un lēmumu punktus.

5.attēls. Stāvokļu diagramma pasūtījuma veikšanai

6.attēls. Stāvokļu diagramma autorizēšanai

## 4.3. Datu struktūru apraksts

Datu bāzes shēma parāda, kā dažādas tabulas ir saistītas savā starpā, lai uzglabātu un pārvaldītu datus par rotājumiem, lietotājiem, pasūtījumiem un citiem veikala elementiem.

Šeit ir detalizēts apraksts par katru tabulu:

1. **lietotaji\_sparkly** - Lietotāju datu tabula

* Glabā informāciju par sistēmas lietotājiem (klientiem, administratoriem, moderatoriem)
* Satur laukus: id\_lietotajs (primārā atslēga), vards, uzvards, lietotajvards, epasts, parole, foto, loma, datums
* Loma norāda lietotāja tiesību līmeni (klients, administrators vai moderators)
* Datubāzē saglabā lietotāja reģistrācijas laiku

1. **grozs\_sparkly** - Pirkuma groza tabula

* Uzglabā informāciju par lietotāju iepirkuma groziem
* Satur laukus: id\_grozs (primārā atslēga), bumba\_id, daudzums, statuss, datums
* Statuss var būt "enum" tipa, norādot groza statusu (piemēram, aktīvs, pasūtīts)
* Saistīta ar lietotāju tabulu, kas norāda, kuram lietotājam pieder grozs

1. **produkcija\_sparkly** - Produktu tabula

* Centrālā tabula, kas satur informāciju par visiem pieejamajiem rotājumiem
* Satur laukus: id\_bumba (primārā atslēga), forma, nosaukums, audums, malu\_figura, dekorejums1, dekorejums2, attels1, attels2, attels3, cena, daudzums, ipass\_pied, timestamp
* Saistīta ar vairākām citām tabulām, kas uzglabā informāciju par dažādām rotājumu sastāvdaļām
* Glabā arī produkta attēlus, cenu un pieejamo daudzumu

1. **sparkly\_formas** - Rotājumu formu tabula

* Satur pieejamo rotājumu formu sarakstu
* Lauki: id\_forma (primārā atslēga), forma (formas nosaukums), datums
* Saistīta ar produkcijas tabulu, lai norādītu, kādas formas ir pieejamas rotājumiem

1. **sparkly\_dekorejums1 un sparkly\_dekorejums2** - Dekorējumu tabulas

* Uzglabā informāciju par pieejamajiem dekorējumiem
* Abas tabulas satur līdzīgus laukus: id, nosaukums, attels, datums
* Saistītas ar produkcijas tabulu, lai norādītu, kādus dekorējumus var izmantot rotājumiem

1. **sparkly\_malu\_figura** - Mālu figūru tabula

* Satur datus par pieejamajām mālu figūrām (dekoratīvajiem elementiem)
* Lauki: id\_malu\_figura, nosaukums, attels, datums
* Saistīta ar produkcijas tabulu, nodrošinot mālu figūru izvēli rotājumiem

1. **sparkly\_audums** - Audumu tabula

* Uzglabā informāciju par pieejamajiem auduma veidiem
* Lauki: id\_audums, nosaukums, datums
* Saistīta ar produkcijas tabulu, nodrošinot auduma izvēli rotājumiem

1. **sparkly\_pasutijumi** - Pasūtījumu tabula

* Satur informāciju par klientu veiktajiem pasūtījumiem
* Lauki: id\_pasutijums (primārā atslēga), lietotajs\_id, kopeja\_cena, apmaksas\_veids, produktu\_skaits, vards, uzvards, epasts, talrunis, pilseta, adrese, pasta\_indeks, piezimes, statuss, pas\_datums, sanemsanas\_datums
* Saistīta ar lietotāju tabulu, lai norādītu lietotāja pasūtījuma informāciju

1. **sparkly\_pasutijuma\_vienumi** – Pasūtījuma vienumu tabula

* Satur detalizēto informāciju par pasūtījuma vienumiem
* Lauki: vienums\_id (primārā atslēga), pasutijuma\_id, produkta\_id, daudzums\_no\_groza, cena, pas\_datums
* Saistīta ar pasūtījumu tabulu, lai parādītu vienumus, kas atrodas pasūtījumā

Šī datubāzes struktūra nodrošina "Sparkly Dream" internetveikala darbību, ļaujot lietotājiem pārlūkot un iegādāties esošos rotājumus, kā arī personalizēt savus rotājumus, izvēloties dažādas formas, dekorējumus un mālu figūras. Struktūra atbalsta arī groza funkcionalitāti un pasūtījumu pārvaldību, kā arī lietotāju autentifikāciju un lomu pārvaldību.

# Lietotāju ceļvedis

## 5.1. Klienta ceļvedis

### 5.1.1. Ielogošanās

* Navigācijas joslā jānospiež poga “”.
* A screenshot of a login form

  AI-generated content may be incorrect.Jāaizpilda lauki: “Lietotājvārds”, “Parole”. (skat. 7.att.)

7.attēls. Autorizēšanās

* Jānospiež poga “Ielogoties”.

### 5.1.2. Reģistrācija

* Ielogošanās lapā (skat. 7.att.) jānospiež poga “Reģistrēties”.
* Jāaizpilda lauki: “E-pasts”, “Vārds”, “Uzvārds”, “Lietotājvārds”, “Parole”, “Parole atkārtoti”. (skat. 8.att.)

A screenshot of a phone

AI-generated content may be incorrect.8.attēls. Reģistrācija

* A blue background with white text

  AI-generated content may be incorrect.Laukiem “Parole” un “Parole atkārtoti” jābūt vienādām, lai rādītos zaļš apstiprinājuma ķeksītis. (skat. 9.att.)

9.attēls. Paroles sakritība

* Jānospiež poga “Reģistrēties”.
* Ja visi dati ir kārtībā, tad lietotājs tiks novirzīts atpakaļ uz ielogošanas lapu, tad ir jāielogojas.

### 5.1.3. Izlogošanās

* Lai izlogoties, navigācijas joslā jānospiež poga “”.

### 5.1.4. Produkta pievienošana grozā

* Lai pievienotu produktu grozā, lietotājam jābūt ielogotam.
* Uzklikšķinot uz konkrēto produktu, atvērās lapa, kur var apskatīt sīkāku informāciju.
* Lai pievienotu produktu grozā, jānospiež poga “A black rectangle with white text

  AI-generated content may be incorrect.” .
* Pēc veiksmīgas produkta pievienošanas grozā, tiks saņemts apstiprinājuma ziņojums

“A green check mark and green circle with black text

AI-generated content may be incorrect.”, ka arī navigācijas joslā pie groza pogas, izmainīsies cipars “A white shopping cart with a blue circle

AI-generated content may be incorrect.”.

### 5.1.5. Produkta daudzuma rediģēšana grozā

* Lai pāriet uz grozu, navigācijas joslā jānospiež poga “A screen shot of a shopping cart

  AI-generated content may be incorrect.” .
* Lai mainītu produkta daudzumu grozā, jānospiež “-” vai “+” pie attiecīga produkta. (skat. 10.att.)

A screenshot of a computer

AI-generated content may be incorrect.10.attēls. Grozs

### 5.1.5. Produkta dzēšana no groza

* Lai dzēstu produktu no groza, jānospiež poga “” pie attiecīga produkta.

### 5.1.6. Groza tīrīšana

* Lai iztīrīt visu grozu reizē, jānospiež poga “”.

### 5.1.7. Pasūtījuma noformēšana

* Lai noformēt pasūtījumu, jānospiež poga “Pasūtīt”. (skat. 10.att.)
* Tiks atvērta jauna lapa ar pasūtījuma kopsavilkumu un formu, kas ir jāaizpild.
* A screenshot of a computer

  AI-generated content may be incorrect.Pēc veiksmīgas forma aizpildes, saņemsiet paziņojumu, kā ari būs iespēja apskatīt pasūtījumu informāciju. (skat. 12.att)

12.attēls. Veiksmīga pasūtījuma noformēšana

### 5.1.8. Personas datu rediģēšana

* Lai apskatīt savu profilu, lietotājam jābūt ielogotam.
* Navigācijas joslā jānospiež poga “”, kur vārds ir lietotāja lietotājvārds.
* Sadaļā “Personas dati” ievadot visus laukus, ko vēlās rediģēt, vai augšupielādējot profila attēlu, jānospiež poga “” .

### 5.1.9. Paroles maiņa vai konta dzēšana

* Lai apskatīt savu profilu, lietotājam jābūt ielogotam.
* Navigācijas joslā jānospiež poga “”, kur vārds ir lietotāja lietotājvārds.
* Sadaļā “Iestātījumi” ievadot veco paroli, jauno paroli un atkārtoti jaunu paroli, jānospiež poga “” .
* Ja vēlās dzēst kontu, jānospiež poga “”, pēc tā ir jāapstiprina savas darbības, atkal nospiežot uz pogu “A black and white sign with white text

  AI-generated content may be incorrect.”.

### 5.1.10. Saziņa ar organizāciju

* Navigācijas joslā jāizvēlās sadaļa “Kontakti”.
* Jāaizpilda forma, ievērojot obligātos laukus, kas ir norādīti ar “\*”, un jānospiež poga “”.

# Testēšanas dokumentācija

## 6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums

Šī projekta testēšanai tika izvēlēta melnās kastes (black box) testēšanas metode ar manuālo testēšanas pieeju. Melnās kastes testēšana koncentrējas uz programmatūras funkcionalitātes pārbaudi no lietotāja perspektīvas, neiedziļinoties iekšējā kodā. Šī pieeja ir ideāli piemērota "Sparkly Dream" veikala testēšanai, jo tā ļauj efektīvi pārbaudīt lietotāja pieredzi un funkcionalitāti, kas ir kritiski svarīga e-komercijas platformai.

Manuālā testēšana - Sistēmas pārbaude tiek veikta, cilvēka testerim imitējot reāla lietotāja darbības un analizējot platformas reakciju. Testeris izpilda noteiktas darbību secības (testpiemērus) un salīdzina faktiskos rezultātus ar sagaidāmajiem rezultātiem.

Manuālā melnās kastes testēšanas metode tika izvēlēta šādu iemeslu dēļ:

Fokuss uz lietotāja pieredzi: Tā kā projekts ir e-komercijas platforma, lietotāja pieredze ir ārkārtīgi svarīga. Manuālā testēšana ļauj novērtēt sistēmu no lietotāja skatupunkta un atklāt problēmas, ko automatizētie testi varētu nepamanīt.

Elastība: Manuālā testēšana ļauj ātri pielāgoties izmaiņām prasībās vai funkcionalitātē, kas ir īpaši svarīgi projekta agrīnajās stadijās.

Intuitīva problēmu identifikācija: Cilvēka testeris var intuitīvi identificēt lietojamības problēmas un dizaina neatbilstības, kas automatizētajiem testiem būtu grūti atklāt.

Pieejamība: Manuālo testēšanu var veikt bez specializētām programmēšanas zināšanām, kas ļauj testēšanā iesaistīt arī ne-tehniskos komandas locekļus.

Manuālās testēšanas procesā tiek izveidoti testpiemēri, kas aptver visas būtiskākās lietotāja darbības sistēmā - no reģistrācijas un autorizācijas līdz produktu apskatei, grozu pārvaldībai un pasūtījumu veikšanai. Testēšanas rezultāti tiek dokumentēti testēšanas žurnālā, kas ļauj sekot līdzi atrastajām problēmām un to novēršanai.

## Alternatīvās testēšanas metodes un rīki

**Baltās kastes (White Box) testēšana**

Baltās kastes testēšana ir metode, kurā tiek pārbaudīta programmatūras iekšējā struktūra, dizains un kods. Šīs metodes ietvaros testeris pārzina programmas iekšējo uzbūvi un var testēt specifiskas koda daļas, funkcijas vai moduļus.

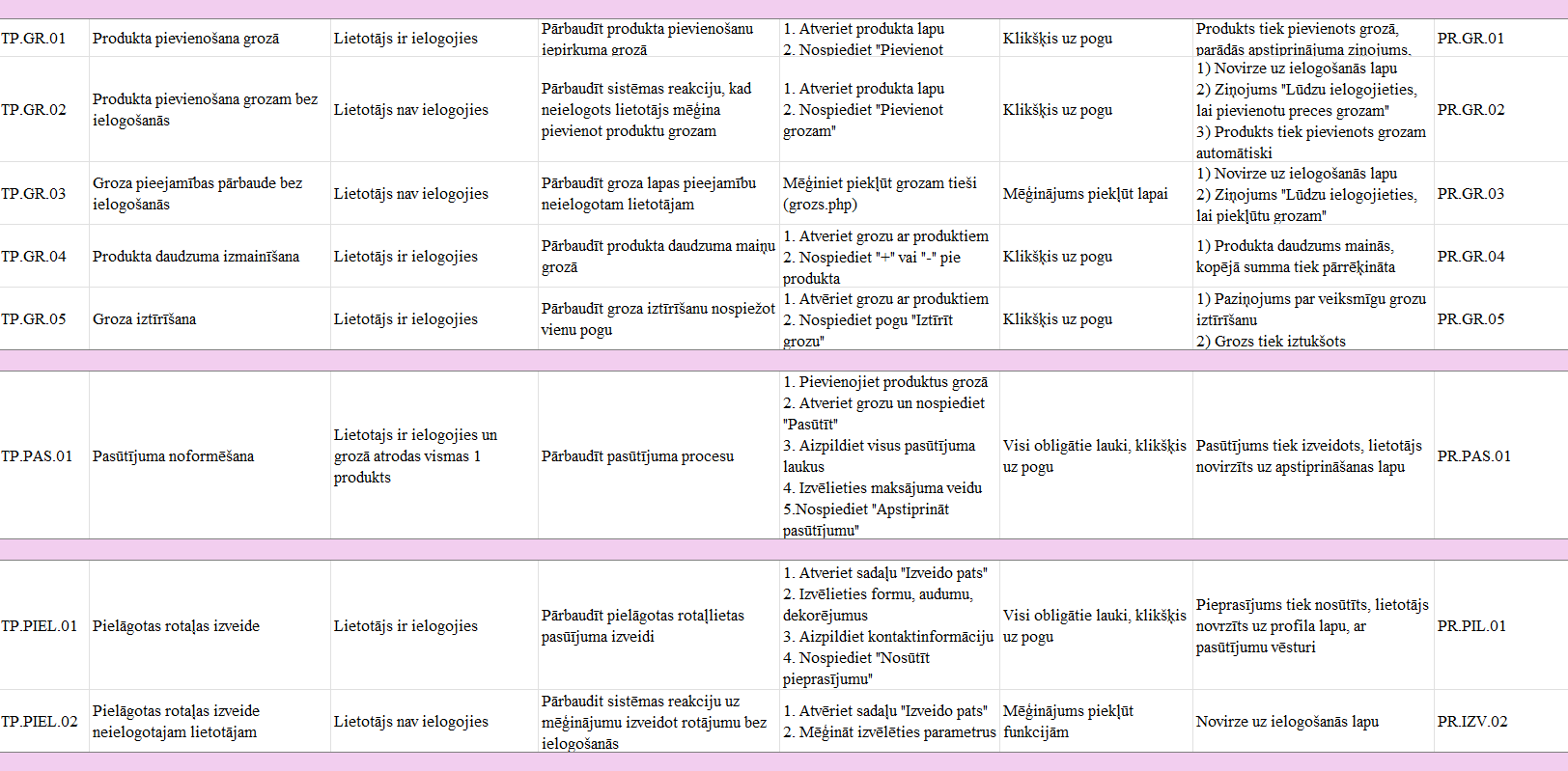
Laika ierobežojumi: Baltās kastes testēšana ir laikietilpīgāka, jo prasa detalizētu koda analīzi un testpiemēru izstrādi katrai koda daļai. Projekta izpildes grafiks neļāva atvēlēt pietiekami daudz laika šādai pieejai.

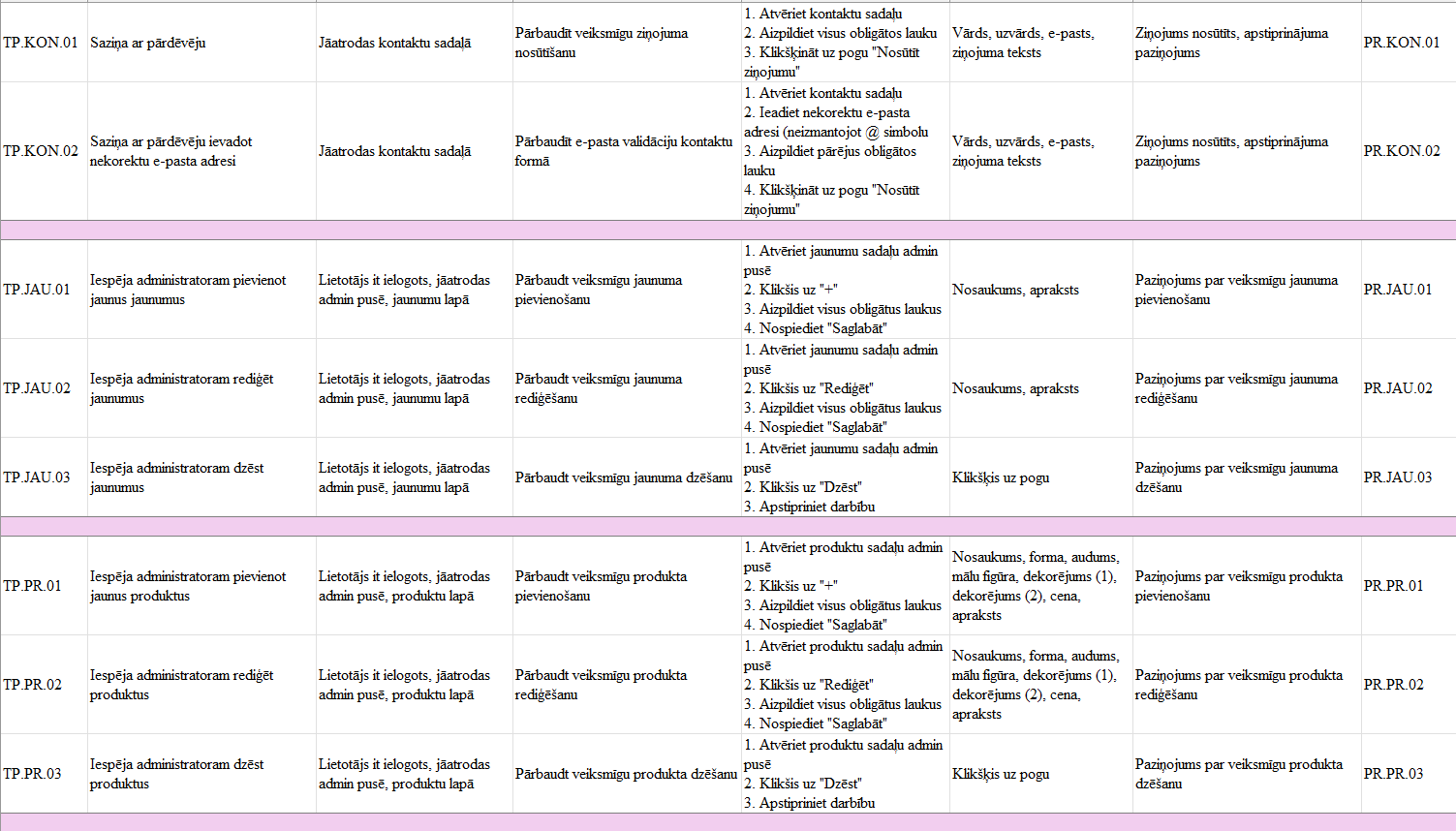
Fokuss uz lietotāja prasībām: Projekta prioritāte bija pārliecināties, ka sistēma atbilst lietotāja prasībām un nodrošina labu lietotāja pieredzi, nevis koda iekšējās struktūras pārbaude.

Projekta apjoms: "Sparkly Dream" ir salīdzinoši neliels projekts ar skaidri definētām funkcionalitātēm, tāpēc manuālā melnās kastes testēšana bija pietiekama, lai nodrošinātu sistēmas kvalitāti.

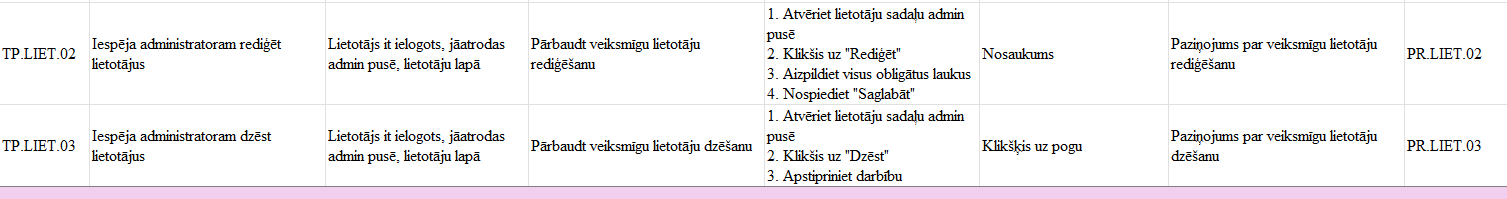
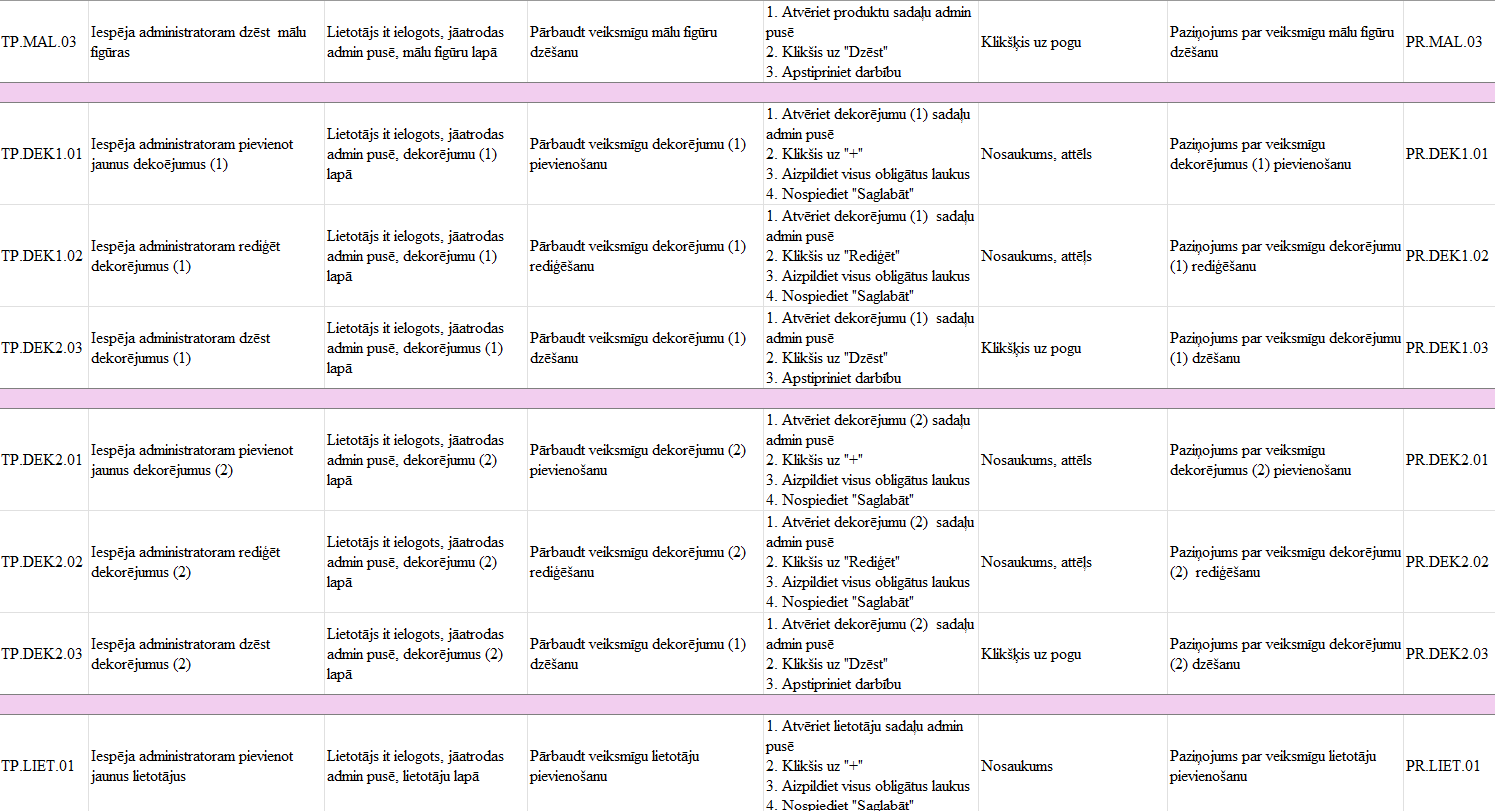
## 6.3. Testpiemēru kopa

# 

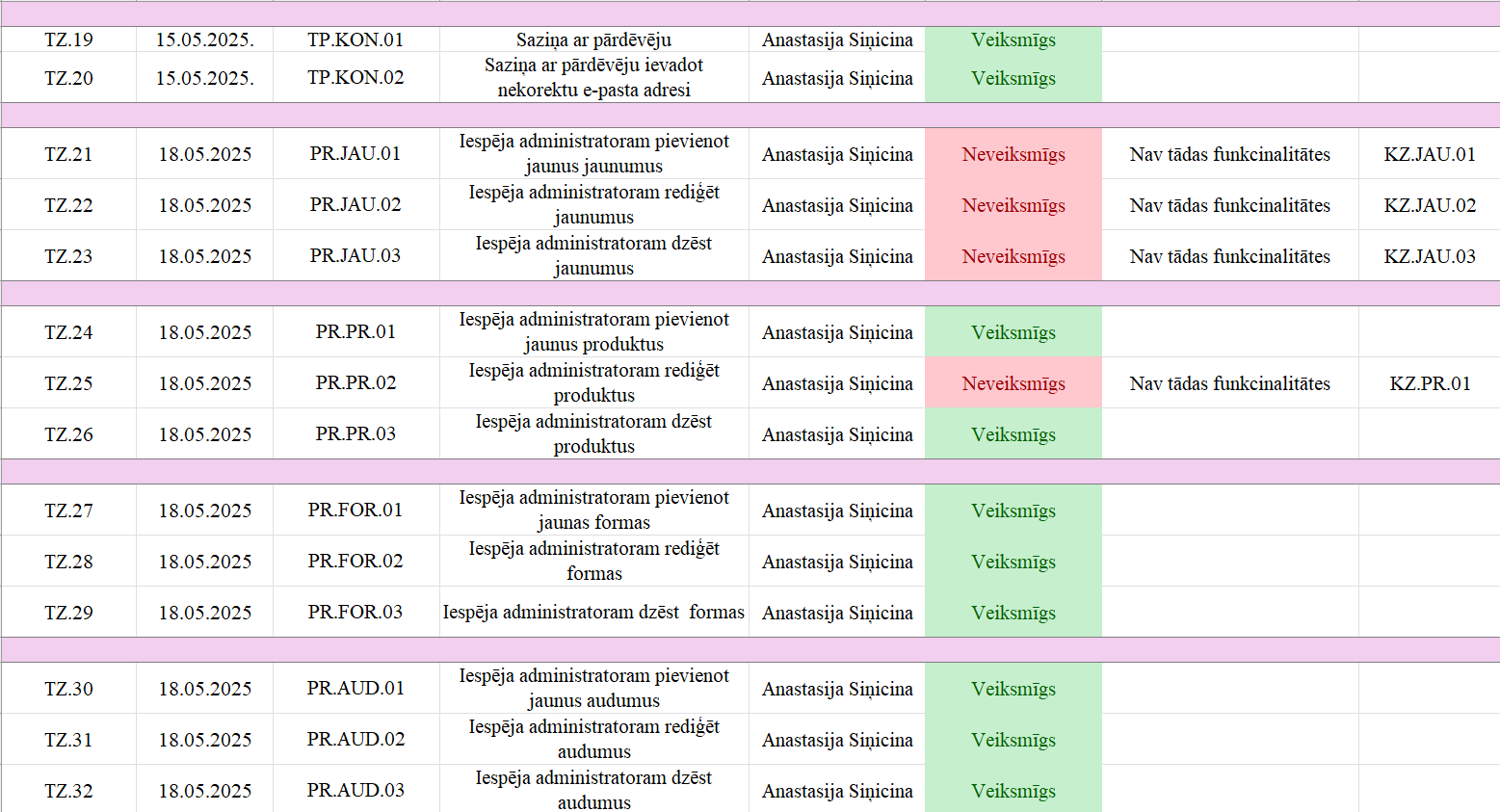
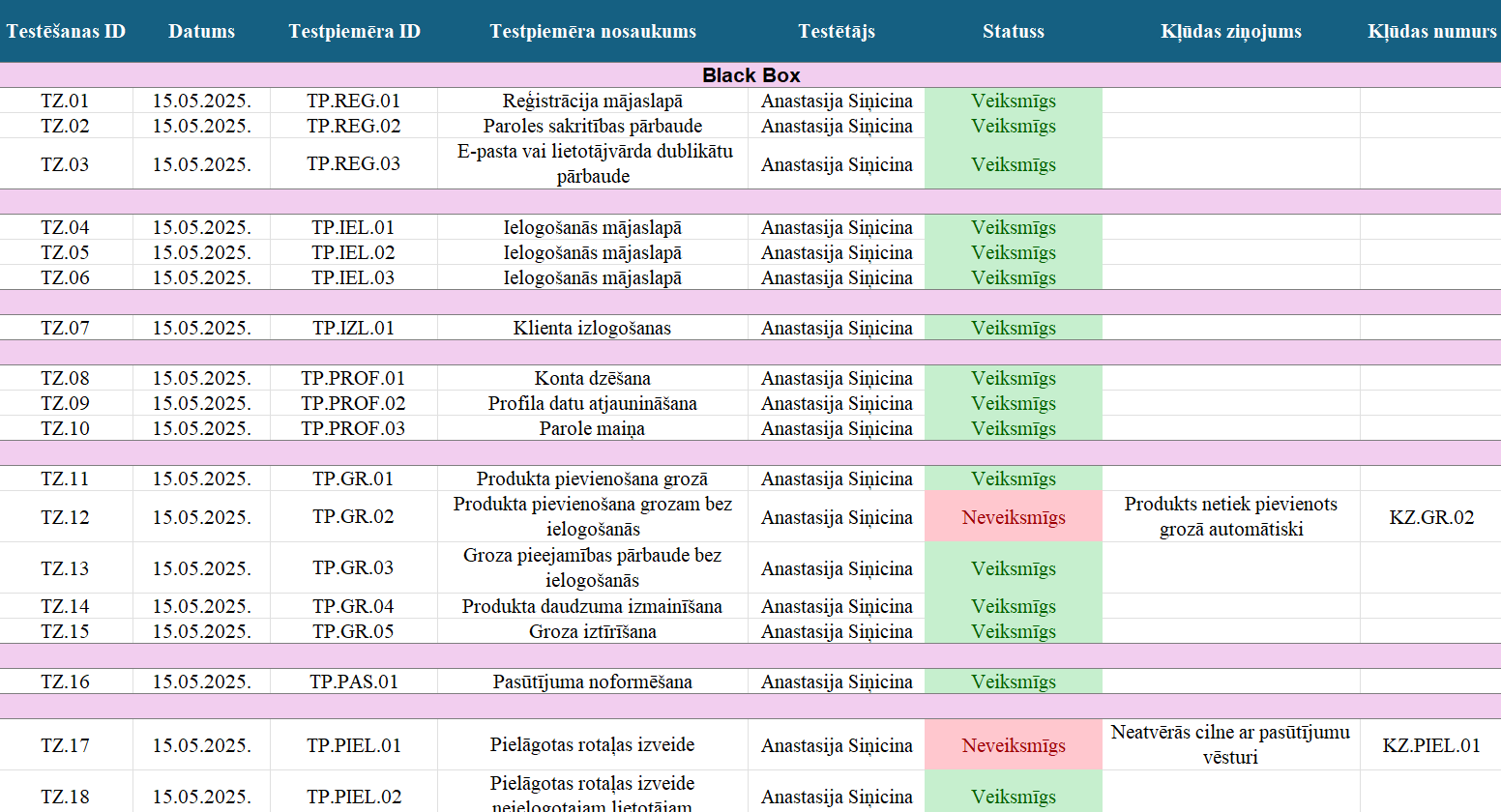


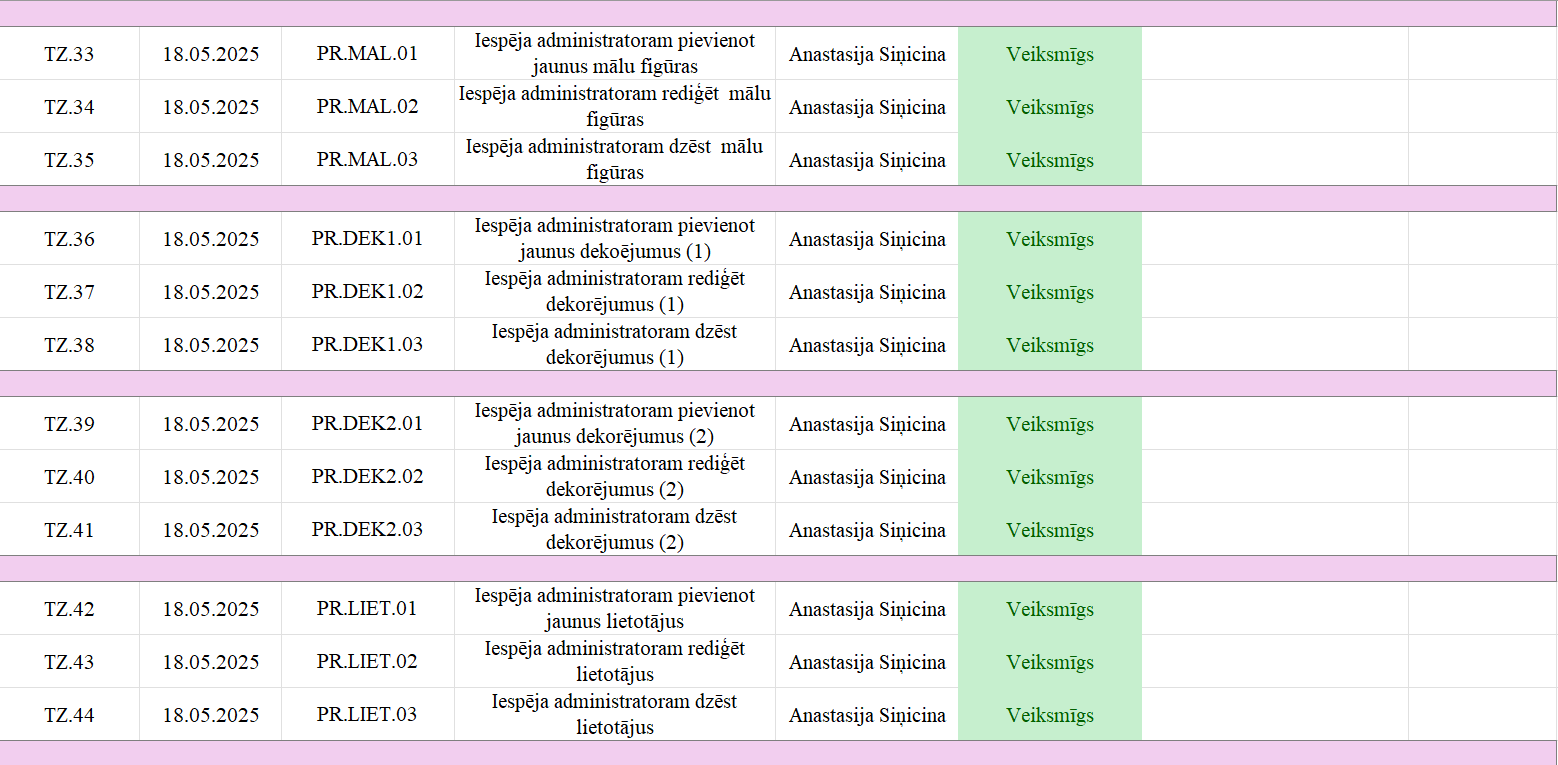






## 6.4. Testēšanas žurnāls





# Secinājumi

Internetveikala "Sparkly Dream" izstrādes process ir bijis vērtīga pieredze, kas ļāvusi apvienot teorētiskās zināšanas ar praktisko pielietojumu reālā biznesa risinājumā. Projekta izstrādes gaitā tika sasniegti vairāki nozīmīgi rezultāti.

Izstrādes procesā tika konstatēti arī daži izaicinājumi, piemēram, nepieciešamība nodrošināt responsīvu dizainu dažādām ierīcēm un autentifikācijas sistēmas drošības jautājumi. Tomēr šie izaicinājumi tika veiksmīgi pārvarēti, un rezultātā izveidota stabila un funkcionāla platforma.

Nākotnē būtu vērtīgi paplašināt sistēmu ar papildu funkcionalitāti, piemēram, papildinātās realitātes elementiem, kas ļautu lietotājiem vizualizēt personalizētos rotājumus, vai daudzvalodu atbalstu, lai paplašinātu potenciālo klientu loku. Tāpat būtu lietderīgi implementēt papildu analītikas rīkus biznesa procesu optimizācijai un klientu uzvedības izpētei.

Kopumā projektā izvirzītie mērķi ir sasniegti, un izveidotā "Sparkly Dream" platforma nodrošina stabilu pamatu ģimenes uzņēmuma izaugsmei digitālajā vidē, vienlaikus kalpojot kā vērtīgs praktiskās pieredzes avots programmatūras izstrādē.

# Lietoto saīsinājumu un terminu skaidrojums

|  |  |
| --- | --- |
| **Saisinājumi un termini** | **Skaidrojums** |
| API | Lietojumprogrammas saskarne, kas ļauj dažādām programmatūras komponentēm savstarpēji komunicēt. |
| AJAX | Tehnoloģija asinhronu tīmekļa lapu atjauninājumu veikšanai, kas ļauj atjaunināt daļu lapas, nepārlādējot visu lapu. |
| CRUD (Create, Read, Update, Delete) | Četras pamata datu manipulācijas funkcijas: izveidošana, lasīšana, rediģēšana un dzēšana. |
| IDE (Integrated Development Environment) | Integrētā izstrādes vide, programmatūras komplekss, kas nodrošina ērtu programmu izstrādi. |
| Frontend | Lietotāja saskarne, kas ir redzama lietotājam un ar kuru tas mijiedarbojas. |
| Backend | Serveris, kas apstrādā datus un nodrošina funkcionalitāti, kas nav tieši redzama lietotājam. |

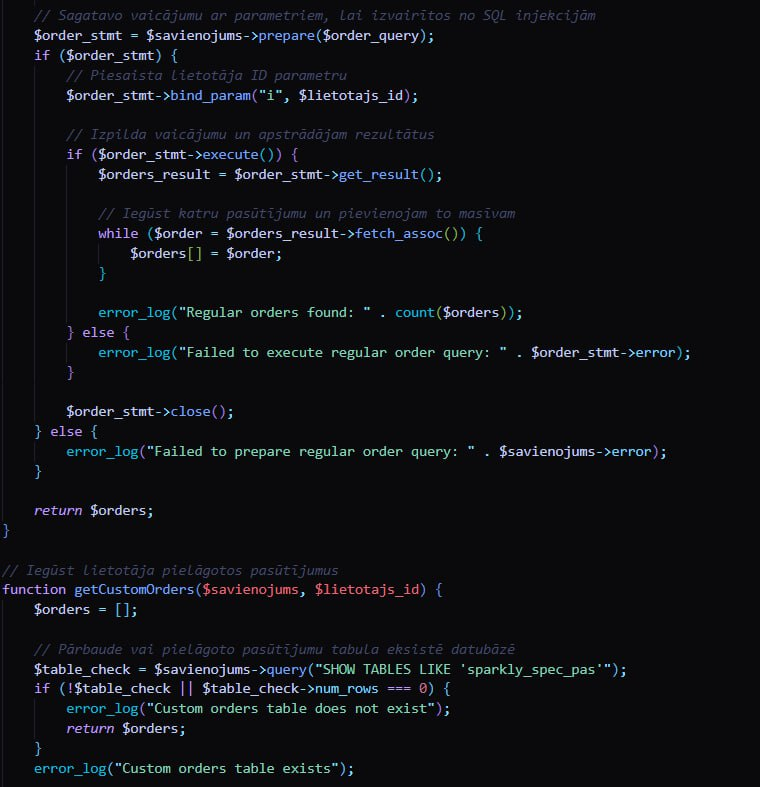
# Literatūras un informācijas avotu saraksts

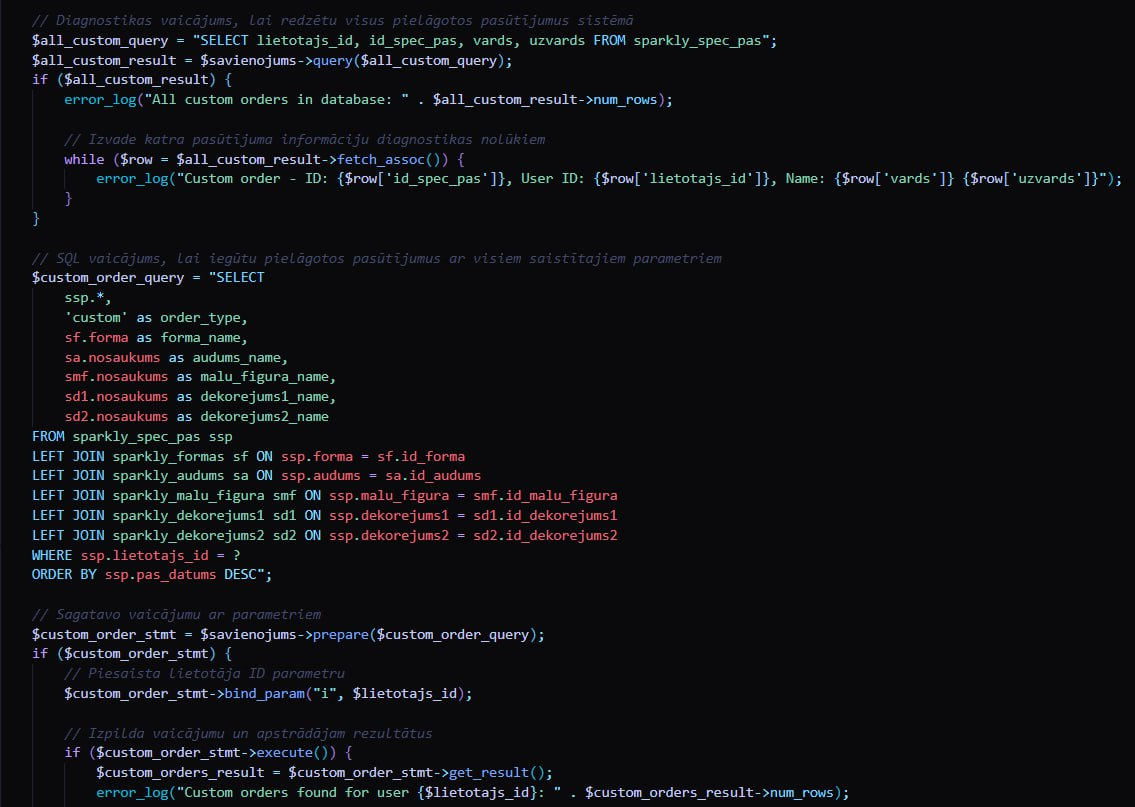
1. W3Schools. (2025). CSS Tutorial. https://www.w3schools.com/css/
2. MDN Web Docs. (2025). CSS: Cascading Style Sheets. https://developer.mozilla.org/en-US/docs/Web/CSS
3. Smashing Magazine. (2024) https://www.smashingmagazine.com/2024/05/ecommerce-ux-design-trends/
4. Nielsen Norman Group. (2024). E-Commerce User Experience. https://www.nngroup.com/articles/ecommerce-ux/

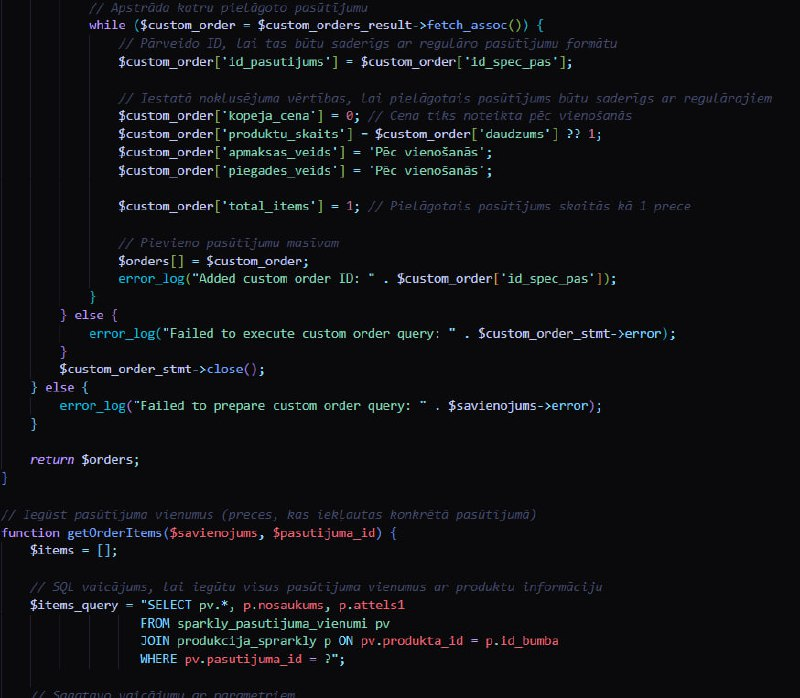
Shopify. (2025). E-commerce Website Structure: Best Practices for Online Stores. <https://www.shopify.com/blog/ecommerce-website-structure>

# Pielikumi

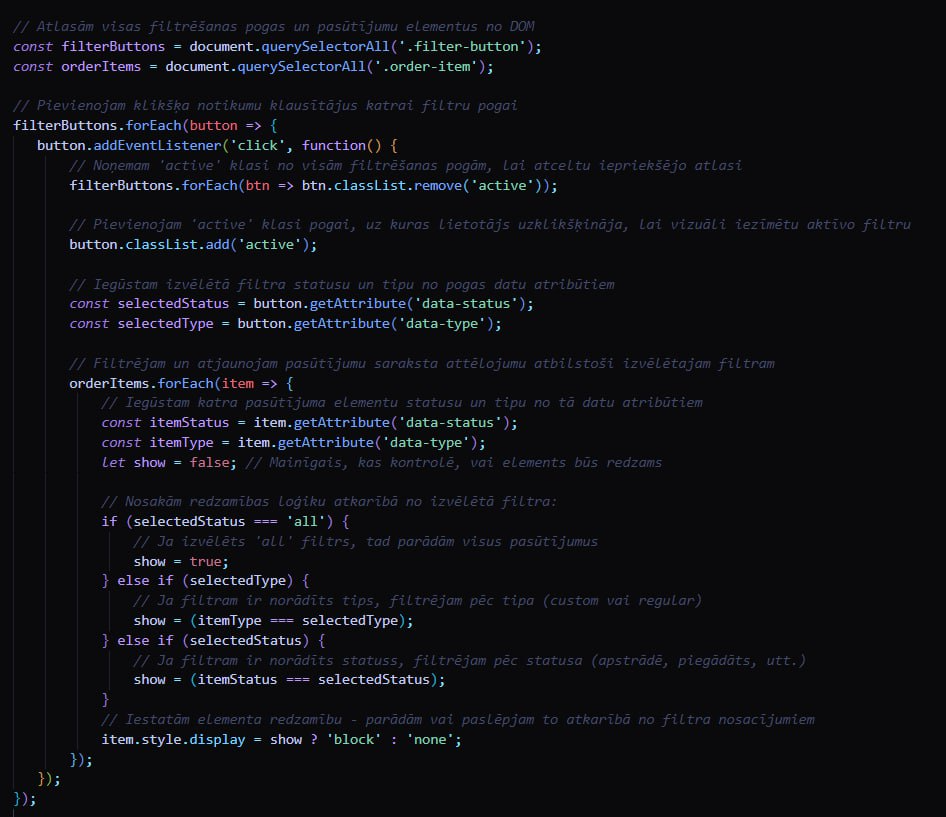
13.attēls. Pasūtījumu apskate un filtrēšana

14.attēls. pasūtījumu apskate un filtrēšana

15.attēls. Pasūtījumu apskate un filtrēšana

16.attēls. pasūtījumu apskate un filtrēšana

17.attēls. pasūtījumu apskate un filtrēšana

18.attēls. Pasūtījumu apskate un filtrēšana