**RĪGAS VALSTS TEHNIKUMS**

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

**KVALIFIKĀCIJAS DARBS**

**“Skaistumkopšanas salona darba uzskaites sistēma”**

Paskaidrojošais raksts 23 lpp.

Audzēknis: Ilona Grebņova

Prakses vadītājs: Ilona Demčenko

Nodaļas vadītājs: Normunds Barbāns

**Rīga**

**2023**

# Anotācija

Šī kvalifikācijas darba mērķis ir sniegt detalizētu izpratni par programmēšanas projektēšanas procesu, iekļaujot prasību specifikāciju, programmatūras produktu modelēšanu un projektēšanu. Darba struktūra ietver vairākas sadaļas, kas sniedz detalizētu izklāstu par šiem procesiem.

Pirmkārt, darbā ir sniegta uzdevuma nostādne, kas iepazīstina lasītāju ar tēmas būtību un darba mērķi. Pēc tam seko prasību specifikācija, kur aprakstītas gan funkcionālās, gan nefunkcionālās prasības, kas ir svarīgas programmēšanas projektam. Šajā sadaļā tiek detalizēti aprakstīta gan ieejas, gan izejas informācija, kā arī konkrētas funkcionalitātes un kvalitātes prasības.

Tālāk darbā tiek aplūkots uzdevuma risināšanas līdzekļu izvēles pamatojums, kur tiek analizētas un izskaidrotas programmēšanas rīku izvēles, piemēram, izstrādes vides un servera izvēli. Tas sniedz pamatojumu un argumentus par to, kāpēc tieši šie rīki tika izvēlēti projektēšanas procesā.

Galvenā daļa darbā ir veltīta programmatūras produkta modelēšanai un projektēšanai. Šajā sadaļā ir detalizēti aprakstīti sistēmas struktūras modelis, sistēmas arhitektūra, ER-modelis, funkcionālais sistēmas modelis un datu plūsmas modelis. Tas sniedz izpratni par sistēmas uzcelšanu, datu plūsmu un attiecībām starp dažādām sistēmas komponentēm un moduļiem.

Pēdējās sadaļās tiek sniegts detalizēts apraksts par datu struktūrām, lietotāja ceļvedi, un nobeigums apkopo galvenās secinājumus un izvirza virzienus turpmākai darbībai. Darba nobeigumā ir arī informācijas avoti un pielikumi, kas papildina un atbalsta darbā izteiktos jautājumus un risinājumus.

Šī kvalifikācijas darba rezultātā ir izveidots pamats turpmākai programmēšanas darbībai, nodrošinot skaidru izpratni par sistēmas struktūru, darbību un funkcionalitāti. Tas sniedz iespēju turpmākai attīstībai un uzlabojumiem programmēšanas projektā.

# Annotation

The aim of this qualification work is to provide a detailed understanding of the programming design process, including requirement specification, software product modeling, and design. The structure of the work includes several sections that provide a detailed overview of these processes.

Firstly, the work provides a statement of the task, introducing the reader to the essence of the topic and the objective of the work. This is followed by the requirement specification, describing both the functional and non-functional requirements that are important for the programming project. This section provides detailed descriptions of both input and output information, as well as specific functionality and quality requirements.

Next, the work examines the justification for the choice of means of solving the task, analyzing and explaining the selection of programming tools, such as development environments and server choices. It provides justification and arguments for why these tools were chosen in the design process.

The main part of the work is dedicated to software product modeling and design. This section provides detailed descriptions of the system's structure model, system architecture, ER model, functional system model, and data flow model. It provides an understanding of the system's construction, data flow, and relationships between various system components and modules.

In the last sections, a detailed description of data structures, user guide, and conclusion are provided, summarizing the main findings and proposing directions for further action. The conclusion of the work includes information sources and appendices that complement and support the issues and solutions raised in the work.

As a result of this qualification work, a foundation for further programming work has been established, providing a clear understanding of the system's structure, operation, and functionality. It provides an opportunity for further development and improvement in the programming project.

Saturs

[Ievads 6](#_Toc7208)

[1. Uzdevuma nostādne 7](#_Toc23838)

[2. Prasību specifikācija 9](#_Toc19770)

[2.1. Ieejas un izejas informācijas apraksts 9](#_Toc9443)

*[2.1.1. Ieejas informācijas apraksts](#_Toc8989)* [9](#_Toc8989)

*[2.1.2. Izejas informācijas apraksts](#_Toc24889)* [10](#_Toc24889)

[2.2. Funkcionālās prasības 10](#_Toc25399)

[2.3. Nefunkcionālās prasības 11](#_Toc13367)

[3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS 15](#_Toc22653)

[4. Programmatūras produkta modelēšana un projektēšana 16](#_Toc4431)

[4.1. Sistēmas struktūras modelis 16](#_Toc29597)

*[4.1.1. Sistēmas arhitektūra](#_Toc27986)* [16](#_Toc27986)

*[4.1.2. Sistēmas ER-modelis](#_Toc6015)* [17](#_Toc6015)

[4.2. Funkcionālais sistēmas modelis 18](#_Toc2712)

*[4.2.1. Datu plūsmu modelis](#_Toc19458)* [18](#_Toc19458)

[5. Datu struktūru apraksts 19](#_Toc8660)

[5.1. Datu bāzes fiziskā struktūra 19](#_Toc21403)

[6. Lietotāja ceļvedis 23](#_Toc24644)

[6.1. Sistēmas prasības aparatūrai un programmatūrai 23](#_Toc24893)

[6.2. Sistēmas instalācija un palaišana 24](#_Toc677)

[6.3. Programmas apraksts 26](#_Toc9702)

[6.4. Testa piemērs 26](#_Toc2151)

[Nobeigums 27](#_Toc5125)

[Informācijas avoti 28](#_Toc20475)

[Pielikumi 29](#_Toc22300)

# Ievads

Mūsdienās ir ļoti svarīgi izmantot tehnoloģijas un inovācijas, lai uzlabotu ikdienas dzīvi un piedāvātu efektīvus risinājumus dažādām vajadzībām. Šajā kontekstā sejas un ķermeņa kopšanas nozare nav izņēmums. Lai pilnveidotu šo nozari un padarītu to pieejamu un efektīvu visiem, ir nepieciešams ieviest modernas, racionālas un pielāgojamas sejas un ķermeņa kopšanas apkalpošanas sistēmas.

Šīs sistēmas galvenais mērķis ir nodrošināt plašu pakalpojumu klāstu, kas ietver ne tikai tradicionālas fiziskās klātienes procedūras, bet arī jaunievedumus, piemēram, iespēju veikt pirkumus tiešsaistē un saņemt konsultācijas attālināti. Šāda integrēta pieeja ne tikai atvieglo klientiem piekļuvi pakalpojumiem, bet arī paplašina uzņēmējdarbības iespējas, nodrošinot plašāku klientu loku un efektīvāku resursu izmantošanu.

Šīs sistēmas interneta veikals ļaus klientiem ērti iegādāties produktus un uzņēmējiem paplašināt tirgus sasniegumus. Konsultācijas klātienē un attālināti nodrošinās personīgu pieeju klientiem, pievienojot vērtību pakalpojumu kvalitātei. Tas veicinās gan klientu apmierinātību, gan uzņēmējdarbības efektivitāti.

Šajā kontekstā izveidotā skaistumkopšanas uzskaites sistēma simbolizē pāreju uz modernu un klientu orientētu pakalpojumu sniegšanu, kur tiek apvienotas tradicionālās vērtības ar digitālajām iespējām, radot visaptverošu un efektīvu piedāvājumu šajā nozarē.

# Uzdevuma nostādne

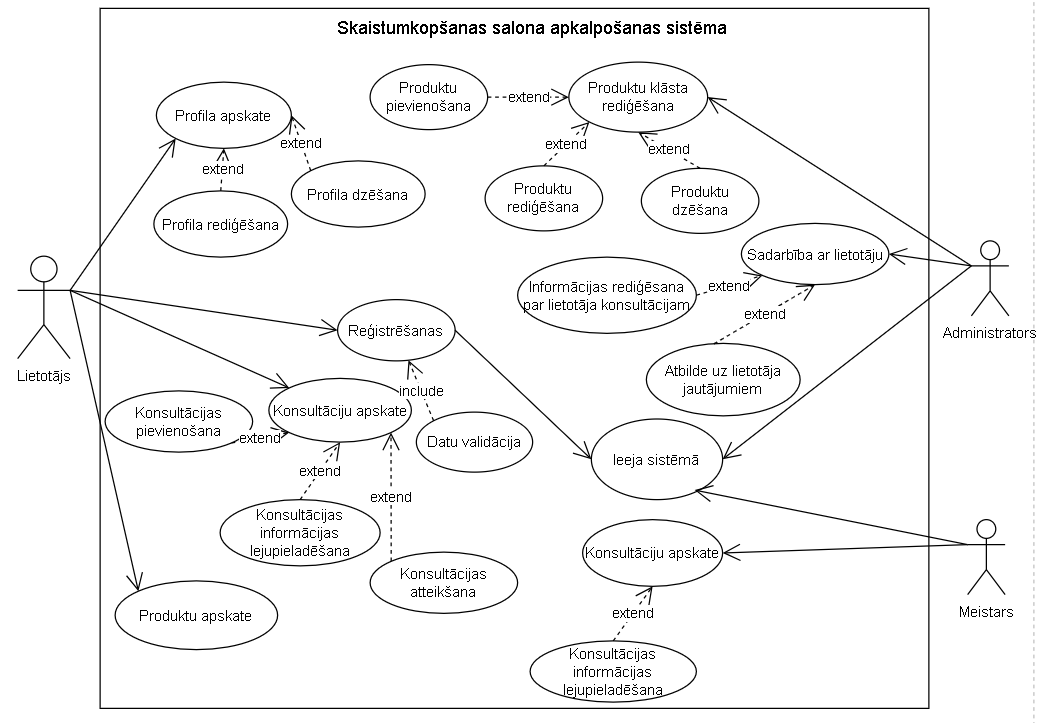
Kvalifikācijas darba uzdevums ir izveidot skaistumkopšanas salona uzskaites sistēmu. Sistēmā nepieciešams realizēt iespēju lietotājiem reģistrēties un pierakstīties, mainīt profila iestatījumus, pievienot, rediģēt un dzēst informāciju par sevi, apskatīt produktu klāstu un pierakstīties uz konsultācijām gan klātienē, gan attālināti. Administratoriem pieejamas tādas funkcijas kā: pārskatīt produktu klāstu, pievienot, rediģēt un dzēst informāciju par produktiem, apskatīt lietotāja profilus, dzēst lietotāju profilu, kā arī mainīt informāciju par lietotāja pierakstiem uz konsultācijām.

Ikvienam ir svarīgi veikt sejas un ķermeņa kopšanu, daudziem tas pat ir tā saucamā "ikdienas rutīna". Šāda tipa tīmekļa vietnes, kur ir pieejams preču katalogs un iespēja pieteikties konsultācijām, nav pilnībā optimizētas sistēmas. Šī projekta mērķis ir izstrādāt tādu sistēmu, kur varētu bez jebkādam grūtībām pieteikties arī attālinātām konsultācijām, lai lietotājs varētu ātrāk uzzināt, kāda tipa produkti priekš sejas vai ķermeņa kopšanas viņam/viņai būs vispiemērotākie.

Zemāk, pirmajā attēla var redzēt sekojošas sistēmas lietojumgadījuma diagrammu (skat 1.1.att.)

Skaistumkopšanas uzskaites sistēmai ir jāizpilda vairākas funkcionalitātes:

* lietotāja reģistrēšana un ielogošanas sistēmā;
* lietotāja profila rediģēšana
* preču apskate, pievienošana un iepirkšanās veikalā;
* iespēja lietotājiem pieteikties konsultācijām klātienē un/vai attālināti, kā arī atteikt konsultāciju;
* iespēja lietotājiem dod savu atsauksmi par produktu.
* administratoriem jārediģē informāciju par konsultācijām, mainīt dienu un laiku, vai dzēst lietotāja konsultāciju;
* administratoriem jārediģē informāciju internetveikalā: pievienot, rediģēt vai dzēst produktus;
* administratoriem jārediģē informāciju par lietotāju: tā profila iestatījumus vai dzēst profilu.



1.1. att. Lietojumgadījuma diagramma

# Prasību specifikācija

## 2.1. Ieejas un izejas informācijas apraksts

### *2.1.1. Ieejas informācijas apraksts*

Sistēmā tiks nodrošināta šādas ieejas informācijas apstrāde.

1. Informācija par **lietotājiem** sastāvēs no šādiem datiem.

* Vārds un uzvārds - lietotāja vārds un uzvārds - burtu teksts ar izmēru līdz 100 rakstzīmēm (piem., Māra Bitte).
* E-pasts - lietotāja reģistrēts vai izmainīts elektroniskais pasts - burtu teksts ar izmēru līdz 100 rakstzīmēm (piem., bittemara@gmail.com).
* Parole - lietotāja reģistrēta vai izmainīta parole - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem., Qwletgd12!).
* Lietotāja attēls - attēls, kuru lietotājs pievienoja sev profilā - izmērs līdz 255 rakstzīmēm.
* Lietotāja tips - apzīmē tipu, lietotājs vai administrators - vesels skaitlis (piem., 0-lietotājs, 1-administrators, 2-meistars).
* Numurs - lietotāja pievienots telefona numurs - teksts ar izmēru līdz 12 rakstzīmēm (piem., +37128392019).

2. Informācija par **produktiem** sastāvēs no šādiem datiem.

* Nosaukums - produkta nosaukums - burtu teksts ar izmēru līdz 255 rakstzīmēm (piem., The Ordinary Glycolic Acid).
* Bilde - produkta bilde - administrators pievienojot produktu arī pievieno tā bildi (piem., jpg formātā).
* Izmērs - produkta lielums - burtu teksts līdz 50 rakstzīmēm (piem., 30ml vai 20g).
* Cena - produkta noteikta cena - daļskaitlis ar precizitāti līdz 2 cipariem aiz komata (piem 6.79).
* Apraksts - produkta detalizēts apraksts - burtu teksts ar izmēru līdz 2000 rakstzīmēm (piem., Eksfoliējošais toniks palīdz cīnīties ar blāvu ādas toni, raupju tekstūru).

3. Informācija par **konsultācijām** sastāvēs no šādiem datiem.

* Pievienošanas datums - datums, kad tika pievienota konsultācija - noformēts datums (piem., 29.10.2023).
* Konsultācijas laiks - laiks kurā notiks konsultācija - iestatīts laiks (piem., 15:00).
* Konsultācijas veids - kāda veidā tiks norisināta konsultācija - burtu teksts ar izmēru līdz 100 rakstzīmēm (piem., Attālināti MS Teams vietnē).

### *2.1.2. Izejas informācijas apraksts*

1. **Lietotāja profila iestatījumi**. Lietotāja profilā rādīsies vārds, e-pasts, bilde. Gadījumā, ja lietotājs rediģēs kādu no iepriekšminētājiem datiem, tie atjaunosies un tas arī parādīsies lietotājam.
2. **Produktu klāsts.** Tiks atspoguļota informācija par pieejamiem produktiem veikalā, to nosaukums, cena, apraksts un izmērs.
3. **Konsultācijas informācija.** Tiks izvadīta informācija par konsultācijas laiku, datumu un tās veidu, lietotāji to varēs apskatīt savā profilā, meistariem arī būs pieejama tā informācija, kā arī tie varēs ielādēt to pdf faila formātā.
4. **Kategorijas.** Administratoram tiks parādīta informācija par kategorijam, pie kurām varēs piesaistīt meistarus, kurus tās pievienos.
5. **Meistari.** Lietotājiem radīsies informācija par meistariem, to pievienos administrators (administratoram arī būs pieejams saraksts ar visiem meistariem kurus tās pievienos).

## 2.2. Funkcionālās prasības

1. Jānodrošina iespēja reģistrēt jaunu lietotāju.

1.1. Lietotājiem jābūt iespējai aizpildīt visus obligātos laukus, piemēram, vārdu, uzvārdu, e-pastu, paroli, attēlu, lietotāja tipu un telefona numuru.

1.2. Ja kāds no obligātajiem laukiem nav aizpildīts vai tie satur nepareizu formātu, lietotājam ir jāsaņem skaidrs paziņojums par nepieciešamību tos koriģēt.

1.3. Sistēmai jāspēj pārbaudīt, vai ievadītais e-pasts jau eksistē sistēmā, lai novērstu dublikātu veidošanos, un jāizvada paziņojums, ja šāds e-pasts jau ir reģistrēts.

1.4. Parolei jāatbilst drošības standartiem, piemēram, jāsatur gan burti, gan cipari, gan speciālie simboli, un jābūt pietiekami garai.

1. Jānodrošina lietotāja autorizācija.
   1. Aktīviem lietotājiem jābūt iespējai autorizēties, ievadot savu e-pastu un paroli.
   2. Ja kāds no laukiem nav ievadīts, izvadīt par to paziņojumu.
2. Atsauksme par konsultāciju.

3.1. Pēc konsultācijas beigām lietotājam jābūt iespējai atstāt atsauksmi par konsultācijas kvalitāti un sniegto informāciju.

3.2. Apskatītās atsauksmes jābūt pieejamām arī citiem lietotājiem, lai tie varētu veidot izpratni par konsultāciju kvalitāti un konsultantu pieredzi.

1. Konsultācijas.

4.1. Lietotājam ir iespēja pieteikt klātienes vai attālināto konsultāciju.

4.2. Gadījumā, ja noteiktā datumā uz noteikto laiku ir pieteikta konsultācija, par to sistēmai jāziņo, lai kāds lietotājs nepieteiktu konsultāciju atkārtoti.

4.3. Ja pēkšņi lietotājs nevar ierasties uz konsultāciju, viņam to ir jāatsaka, vai jāpaziņo administratoram.

5. Lietotāja profils:

5.1. Lietotājam ir jāvar rediģēt savu profilu, pievienojot vai atjauninot informāciju par sevi.

5.2. Ja lietotājs aizmirst paroli, jānodrošina drošs veids, kā to atjaunot vai atgūt.

6. Produktu klāsts:

6.1. Administrātoriem jābūt iespējai pievienot produktus, norādot to nosaukumu, cenu, aprakstu un pieejamo daudzumu.

6.2. Lietotājam jābūt redzami visi pievienoti produkti lapā produktu klāsts.

7. Atsauksmes par produktu.

7.1. Lietotājam ir jābūt iespēja sniegt atsauksmi par produktu.

## 2.3. Nefunkcionālās prasības

1. Sistēmas saskarnes valodai ir jābūt latviešu valodai.

1.1. Visas teksta un interaktīvās elementu uzrādīšanai jābūt latviešu valodā un angļu valodā.

1.2. Lietotājiem jābūt iespējai viegli mainīt saskarnes valodu no sistēmas iestatījumiem.

1. Dizainam ir jābūt saskaņotās krāsās.

2.1. Jāizstrādā krāsu palete, kas atbilst uzņēmuma vai zīmola identitātei.

2.2. Krāsu izmantošanai jābūt harmoniskai un atbilstošai lietotāja pieredzei.

3. Tīmekļa vietnei jābūt pielāgotai visiem ekrāna izmēriem.

3.1. Responsīvās dizaina principi jāievēro visās lapu struktūrās un elementos.

3.2. Jānodrošina, ka vietne tiek pareizi parādīta gan uz lielformāta monitoriem, gan mobilajiem ierīcēm.

4. Lietotāja saskarnei jābūt saderīgai ar populārākajiem interneta pārlūkiem.

4.1. Pārliecināšanās par pilnīgu saderību ar visām galvenajām pārlūkprogrammām, ieskaitot jaunākās versijas.

4.2. Regulāra saderības pārbaude un pielāgošana pēc nepieciešamības.

5. Sistēmai ir jānodrošina uzticama darbība un jānovērš neplānoti traucējumi vai datu zudumi.

5.1. Regulāra dublējumu veidošana un datu atjaunošanas iespējas nodrošināšana.

5.2. Nepārtraukta sistēmas uzraudzība un ātra reaģēšana uz iespējamiem traucējumiem.

6. Ir nepieciešams garantēt datu drošību un privātumu sistēmā.

6.1. Izmantojot end-to-end šifrēšanu, nodrošināt drošu datu pārraidi starp klienta ierīcēm un serveri.

6.2. Regulāri auditēt sistēmas drošības politikas un veikt nepieciešamās uzlabošanas.

1. Sistēmai jābūt ar ātru reakciju un jāgarantē īss atbildes laiks.

7.1. Optimizēt datu apstrādes procesus un servera resursu izmantošanu, lai nodrošinātu maksimāli ātru darbību.

7.2. Izmantot kešatmiņas tehnoloģijas, lai uzlabotu lapu ielādes ātrumu un atbildes laiku.

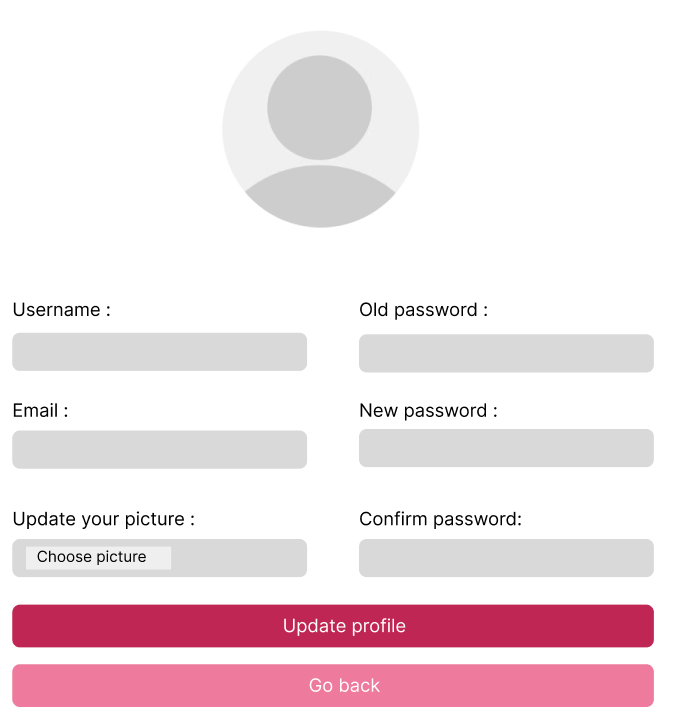
8. Lietotāja pieredze:

8.1. Izmantot lietotāja testēšanu un atsauksmes, lai nepārtraukti uzlabotu lietotāja pieredzi.

8.2. Nodrošināt saprotamu un intuitīvu navigāciju, izmantojot vienkāršas ikonas un vadlīnijas.

Sistēmas ekrānu skices:

* Lietotāja profila skice (skat 2.1. att)

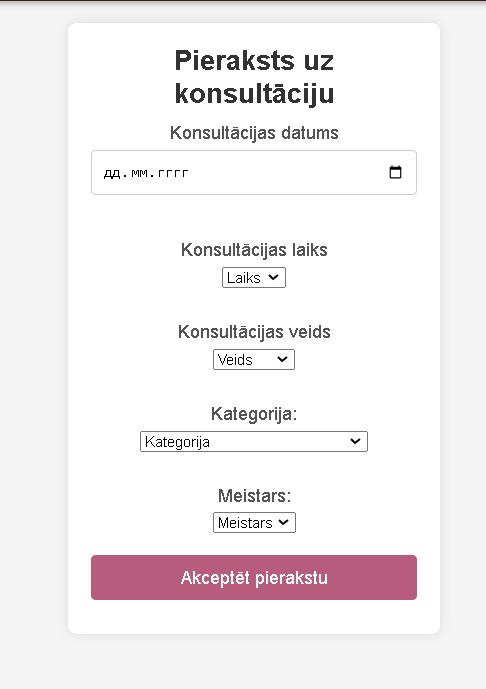
2.1. att. Lietotāja profila skice

* Produktu klāsta skice (skat 2.2. att)



2.2. att. Produktu klāsta skice

* Produktu klāsta skice (skat 2.3. att)



2.3. Konsultācijas pieteikšanas skice

# UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS

Lai nodrošinātu efektīvu un strukturētu pieeju web aplikācijas izveidei, testēšanai un uzturēšanai, ir svarīgi izvēlēties pareizos izstrādes rīkus un vidi. Šajā projektā mēs izmantosim vairākus specializētus rīkus un tehnoloģijas, lai optimizētu izstrādes procesu un nodrošinātu augstas kvalitātes gala rezultātu.

**Visual Studio Code v.1.85** ir izvēlēts kā izstrādes vide, jo tā ir viegli pielāgojama, piedāvā plašu spraudņu atbalstu un efektīvi integrējas ar daudzām programmēšanas valodām, tostarp PHP un JavaScript. Tas ir viegli lietojams un piedāvā ērtu kodu rediģēšanu, atkļūdošanu un projektu pārvaldību.

**XAMPP v.8.2.12** ir izvēlēts kā lokālais serveris un attīstības vide, jo tas nodrošina iespēju lokāli izstrādāt un testēt web aplikācijas. Integrētais Apache serveris un MySQL datubāze ļauj veikt pilnvērtīgu izstrādi, pirms projekts tiek publicēts uz reālo serveri.

**PHP** ir servera puses skriptu valoda, ko bieži izvēlas web izstrādē. Tās lietojums ar XAMPP ļauj veikt dinamisku lapu izveidi un datu apstrādi. Turklāt, PHP var izmantot kā backend valodu, kas mijiedarbojas ar datubāzi un nodrošina dinamisku satura ģenerēšanu.

**CSS** ir izvēlēts, lai veidotu pievilcīgu un labi strukturētu lapas izskatu. Ar tā palīdzību var veikt stilizāciju, pielāgojumu un izkārtojuma iestatījumus, nodrošinot profesionālu un lietotājam draudzīgu interfeisu.

**JavaScript** ir izvēlēts, lai piešķirtu lapai interaktivitāti un dinamiku. Ar JavaScript palīdzību ir iespējams veikt lietotāja ievades apstrādi, dinamiski mainīt lapas saturu un izveidot interaktīvas funkcijas, kas uzlabo lietotāja pieredzi. Tas ir svarīgs elements, lai web aplikācija būtu dinamiska un reaģētu uz lietotāja darbībām.

Kopumā šie rīki nodrošina visaptverošu izstrādes procesu, sākot no lokālā servera un beidzot ar versiju kontroles sistēmu, piedāvājot efektīvu un strukturētu pieeju web aplikācijas izveidei, testēšanai un uzturēšanai.

# Programmatūras produkta modelēšana un projektēšana

## **4.1. Sistēmas struktūras modelis**

### *4.1.1. Sistēmas arhitektūra*

Sistēmā tiek uzskaitīti 3 galvenie moduļi: Lietotājs, kur tiek aprakstīta struktūra gan nereģistrētiem, gan autorizētiem lietotājiem, administrators, kuram ierakstīti viņa uzdevumi un pati sistēma, kurai jānodrošina zēmāk attēlā redzamie uzdevumi.

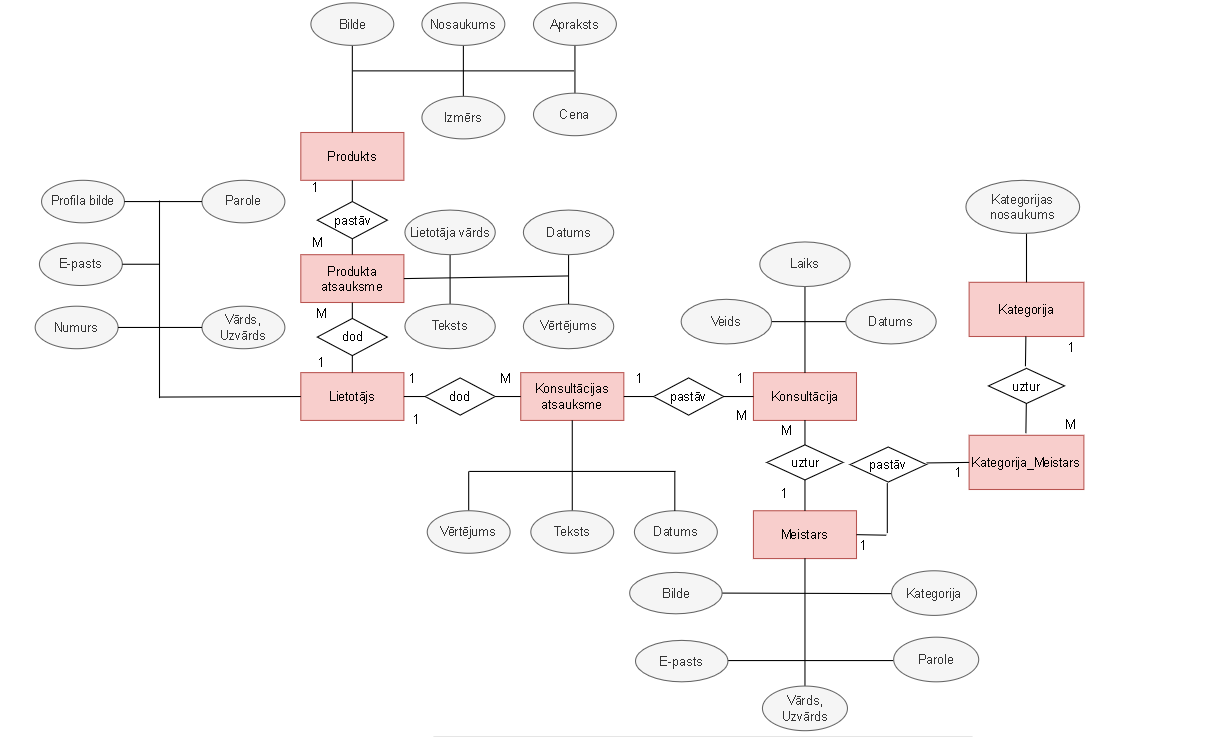
### *chrome_sdqMGFHgor*

4.1. att. Sistēmas funkcionālas dekompozīcijas shēma

* **Lietotāja modulis.** Nereģistrētam lietotājam ir iespēja izveidot jaunu sistēmas lietotāju, apskatīt mājaslapu un produktu klāstu. Reģistrētam lietotājam būs iespēja rediģēt profila bildi, vārdu, e-pastu un paroli, apskatīt produktu klāstu un veikt atsauksmes par konsultācijam/produktiem. Sistēma nodrošinās lietotāju reģistrēšanos caur e-pastu, autorizēšanos un izrakstīšanos. Lietotājam būs iespēja arī pieteikties konsultācijam, informāciju par savām pieteiktām/atceltām konsultācijam tās varēs redzēt savā profilā, kā arī būs iespēja ielādēt par to informāciju pdf faila formātā.
* **Administratora modulis.** Administratorā pusē tiks veiktas izmaiņas par pieejamiem produktiem, konsultācijam, tieši sakot, to rediģēšana, kā arī administratoram būs tiesības apskatīt un pārvaldīt lietotājus.
* **Meistara modulis.** Meistars varēs apskatīt konsultācijas, kuras viņam/viņai ir pievienotas, kā arī lejupielādēt informāciju par to pdf faila formātā.
* **Sistēmas modulis.** Sistēma atjauninās datus par lietotāju profilu rediģēšanu, konsultācijam, konsultāciju/produktu atsauksmēm, produktu rediģēšanas veikšanu, kā arī tiks dzēsta par to informācija.

### ***4.1.2. Sistēmas ER-modelis***

Sistēmas ER-modelis sastāv no 8 entitijām (skat. 4.2. att.), kas nodrošina pamat informācijas uzglabāšanu un apstrādi. Tie ir: lietotājs, produkta atsauksme, produkts, konsultācijas atsauksme, konsultācija, meistars, kategorija\_meistars un kategorija.



4.2. Sistēmas ER-diagramma

Apraksts par galvenām entitātem:

1. Lietotājs:

Lietotājs ir galvenā entītija, kura satur informāciju par vietnē reģistrētajiem lietotājiem. Katram lietotājam ir savs pilnais vārds, e-pasts, parole, bilde.

1. Produkts:

Dota entītija satur informāciju par produktiem kurus pievieno administrators, lietotājs ir iespēja tos visus apskatīt noteiktā lapā. Dati, kas iekļauti šeit: nosaukums, bilde, cena, apraksts un izmērs.

1. Konsultācija:

Entītija satur informāciju par reģistrētam lietotāju konsultācijam, kuras gan lietotāji, gan meistari var apskatīt. Šeit ir iekļauti sekojoši dati: Veids, laiks un datums.

1. Konsultācijas atsauksme:

Šī entītija satur informāciju par lietotāju veiktām atsauksmēm par konsultāciju. Dati, kas iekļauti: vērtējums, teksts, datums.

1. Produkta atsauksme:

Dotā entītija satur informāciju par lietotāju veiktām atsauksmēm par produktiem. Dati, kas šeit tiek iekļauti: lietotāja vārds, vērtējums, teksts, datums.

1. Meistars:

Entītija satur informāciju par meistariem kurus pievienoja administrators. Dati, kuri iekļauti: pilnais vārds, e-pasts, parole, bilde, kategorija.

1. Kategorija\_Meistars:

Dota entītija ir starpsaistoša. Tā saista meistaru un kategoriju, un tāpēc šī entītija uztur informāciju kas tiek ņemta no entītiem meistari un kategorija.

1. Kategorija:

Šī entītija satur informāciju par kategorijam, kuras tiek piesaistītas meistariem. Dati, kas iekļauti: kategorijas nosaukums.

Entītiju saistījums:

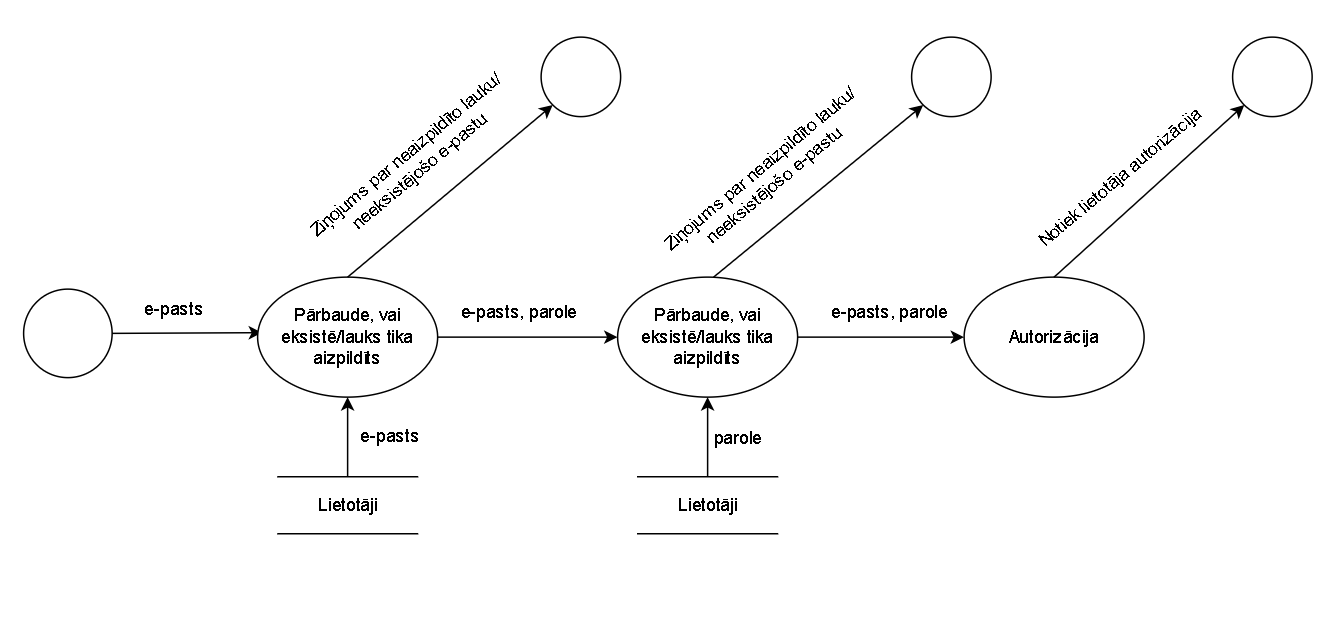
* **Lietotājs (1) - (M) Konsultācijas**, jo viņš var tās pieteikt un pēc konsultācijas atstāt par to atsauksmi. Katrs lietotājs var pieteikt vairākas konsultācijas, un katra konsultācija ir saistīta ar konkrētu lietotāju. Šī saistība ir viens pret daudziem, jo viens lietotājs var reģistrēties uz daudzām konsultācijām. Tas nozīmē, ka lietotājam ir iespēja pieteikt vairākas konsultācijas, piemēram, dažādām dienām un laikiem, atkarībā no viņa vajadzībām un pieejamības.lietotājs (1) ir saistīts ar konsultācijas atsauksmi (M), jo lietotājs var veikt atsauksmi par konsultāciju, kuru novadīja meistars.
* **Lietotājs (1) - (M) Produkta atsauksmes**, jo viņš var veikt atsauksmi par produktu, kuru ir iegādājies. Viens lietotājs var atstāt daudzas atsauksmes par produktiem. Šī saistība ir viens pret daudziem, jo viens lietotājs var atstāt daudz atsauksmju par dažādiem produktiem. Tas nozīmē, ka lietotājs var iegādāties dažādus produktus un atstāt atsauksmes par katru no tiem.meistars (1) ir saistīts ar kategorija\_meistars (1), jo kategorija\_meistars satur informāciju par meistaru.
* **Konsultācija (M) - (1) Meistars**, jo lietotājs izvēlas noteiktu meistaru, lai tas vadītu konsultāciju. Viens meistars var vadīt vairākas konsultācijas. Šī saistība ir daudzi pret vienu, jo daudzas konsultācijas var tikt vadītas ar viena meistara palīdzību. Tas nozīmē, ka viens meistars var nodrošināt daudzas konsultācijas dažādiem lietotājiem dažādos laikos.
* **Lietotājs (1) - (1) Konsultācijas atsauksmes**, jo viņš var atstāt atsauksmi par konsultāciju, kuru ir apmeklējis. Viens lietotājs var atstāt vienu atsauksmi par konsultāciju. Šī saistība ir viens pret vienu, jo viens lietotājs var atstāt vienu atsauksmi. Pēc katras apmeklētās konsultācijas lietotājs var izteikt savu viedokli un novērtējumu.
* **Meistars (M) - (1) Kategorija\_Meistars.** Vairākiem meistariem var būt viena kategorija. Šī saistība ir daudz pret vienam, jo vairāki meistari var būt saistīti ar vienu ierakstu "Kategorija\_Meistars". Tas nozīmē, ka viens ieraksts "Kategorija\_Meistars" norāda uz konkrētu kategoriju un dažādiem meistariem.
* **Kategorija (1) - (M) Kategorija\_Meistars**, jo kategorija\_meistars satur informāciju par kategoriju. Katrai kategorijai var būt daudz ierakstu kategorija\_meistars. Šī saistība ir viens pret daudziem, jo viena kategorija var būt saistīta ar daudziem ierakstiem kategorija\_meistars. Tas nozīmē, kategorija\_meistars norāda uz konkrētu kategoriju, kas ir saistīta ar dažādiem meistariem.
* **Produkts (1) - (M) Produkta atsauksme**. Katram produktam var būt vairākas atsauksmes, bet viena un tā paša atsauksme ir tikai vienam produktam. Šī saite ir viens pret daudziem, jo viens produkts var būt saistīts ar vairākiem pārskatiem.
* **Konsultācijas (1) - (1) Konsultāciju Atsauksmes**. Katrai konsultācijai var būt viena atsauksme. Šī saite ir viens pret vienu, jo viena konsultācija var būt saistīta ar vienu atsauksmi.

## 4.2. Funkcionālais sistēmas modelis

### *4.2.1. Datu plūsmu modelis*

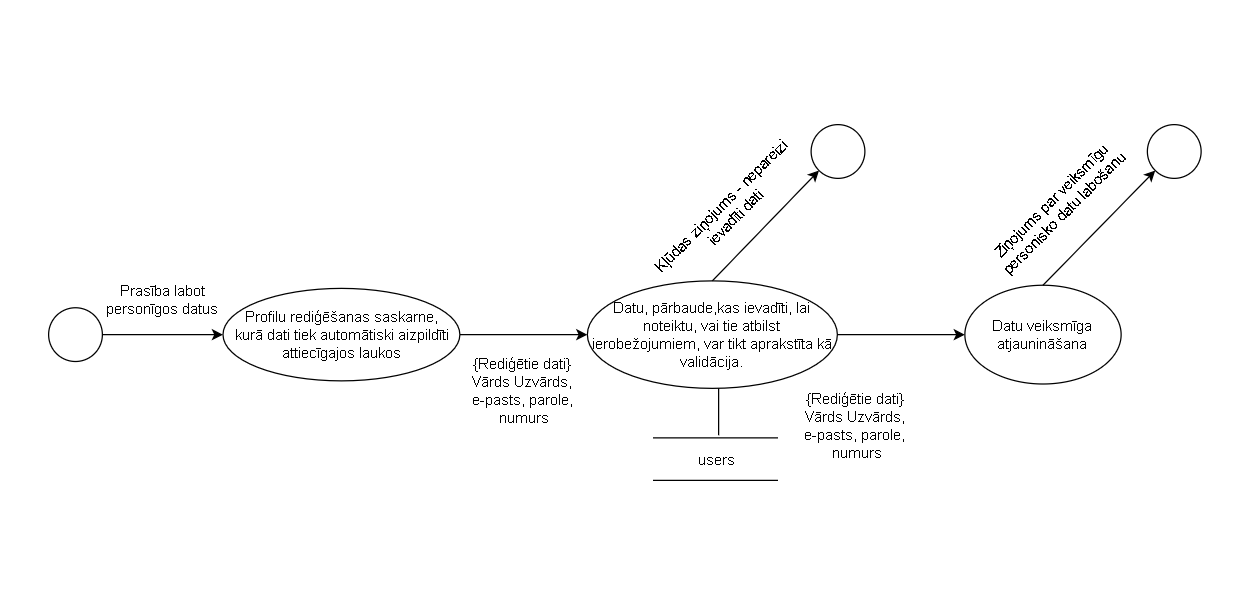
Sistēmā plānots uzturēt šādus datu apstrādes procesus:

1. **autorizācija**: lietotājs ievada e-pasta adresi un paroli. Tiek veikta e-pasta pārbaude. Ja sistēmā sekojošs e-pasts eksistē, tiek pārbaudīta parole ar to, kas pieder kontam. Ja sakrīt gan e-pasts, gan parole, tad sākas autorizācijas sesija. Gadījumā, ja vienā no laukiem bija nepareizi ievadīti dati, tiek izvadīts paziņojums par kļūdu;



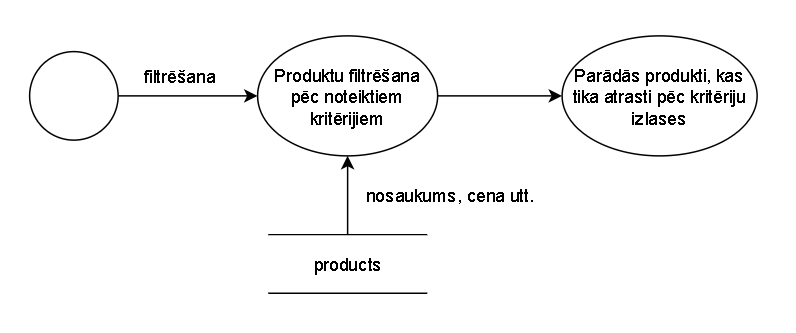
4.3. Datu plūsmas diagramma - lietotāja autorizēšanas

1. **profila rediģēšana**: lietotājam ir iespēja rediģēt nepieciešamo informāciju savā profilā - vārdu, e-pastu, paroli, bildi. Ja lietotājs vecās paroles laukā ievada nepareizu paroli, izvada paziņojumu par kļūdu.



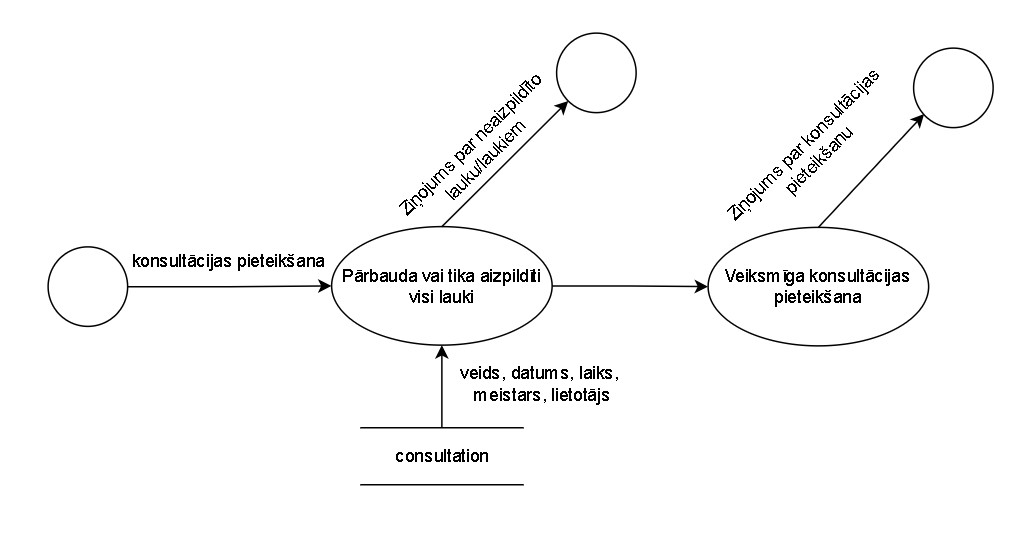
4.4. Datu plūsmas diagramma - lietotāja datu rediģēšana profilā

1. **produktu apskate**: lietotājs var apskatīt produktu klāstu, izmantojot filtrus.

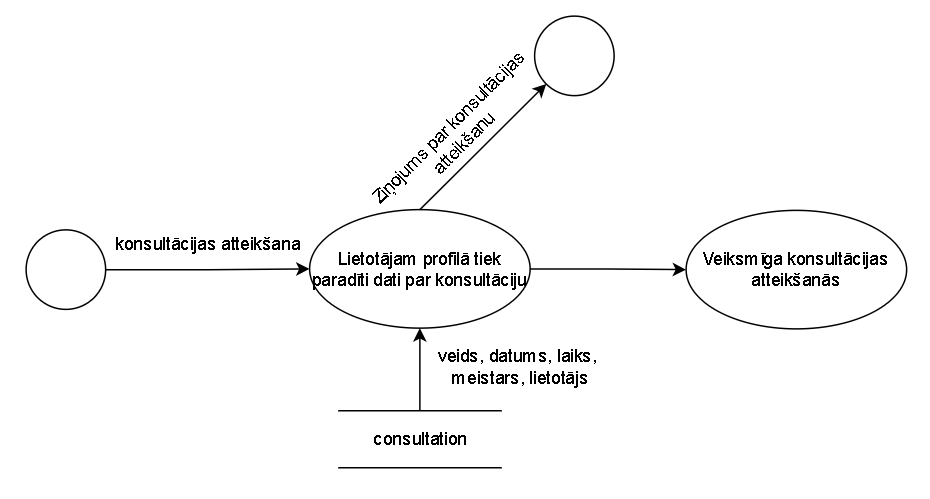


4.5. Datu plūsmas diagramma - produktu filtrēšana pēc kritērijiem

1. **konsultāciju rediģēšana**: lietotājam ir iespēja pievienot konsultāciju uz noteikto datumu un laiku. Ja uz doto datumu un laiku jau tika izveidota konsultācija, lietotājam šīs datums un laiks nerādīsies. Lietotājs arī pats varēs atteikties no konsulāciju, kuru bija pieteicis.

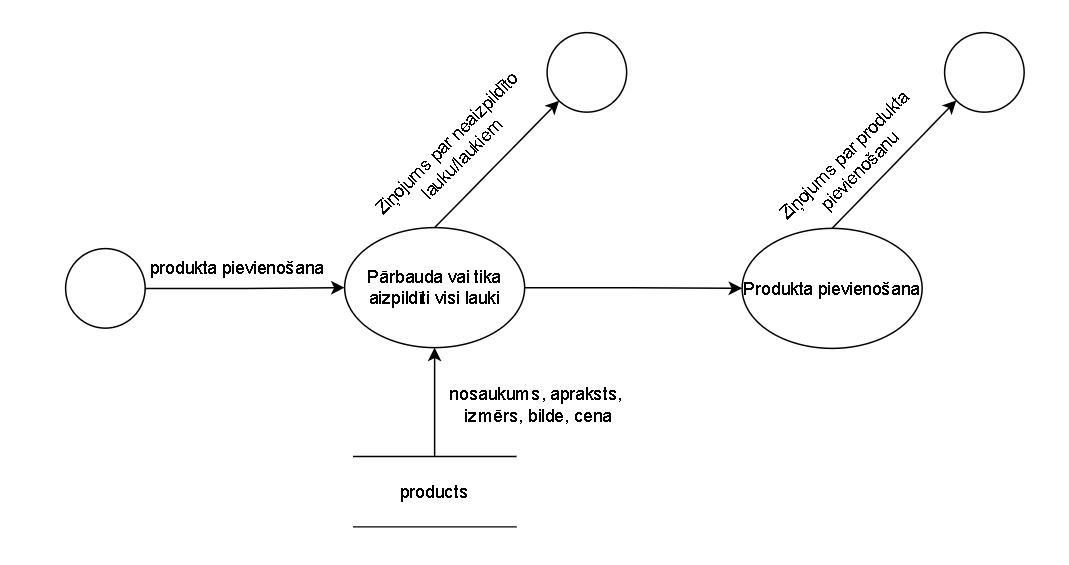
****

4.6. Datu plūsmas diagramma - konsultācijas pieteikšana

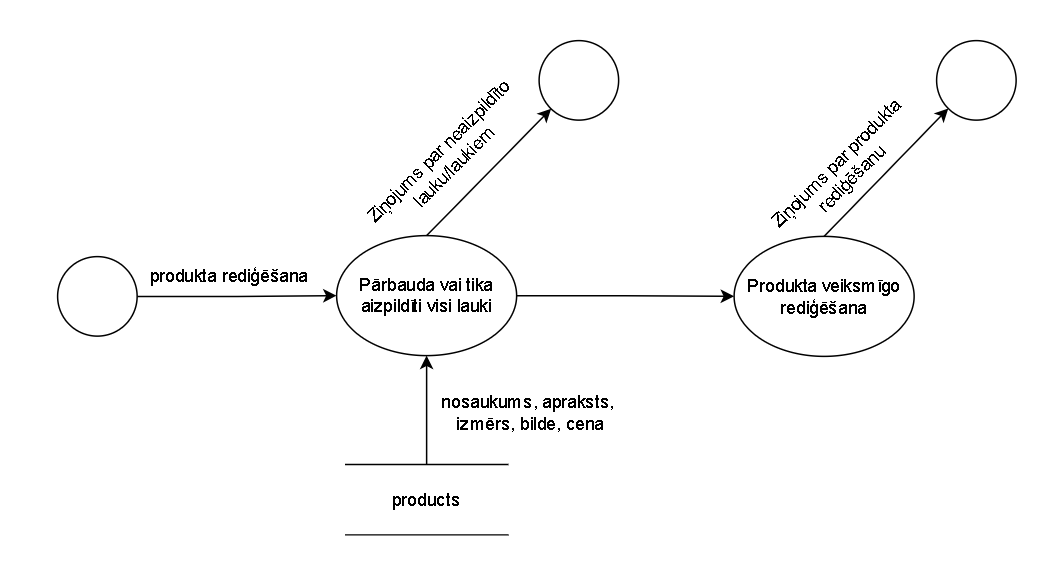
****

4.7. Datu plūsmas diagramma - konsultācijas atteikšana

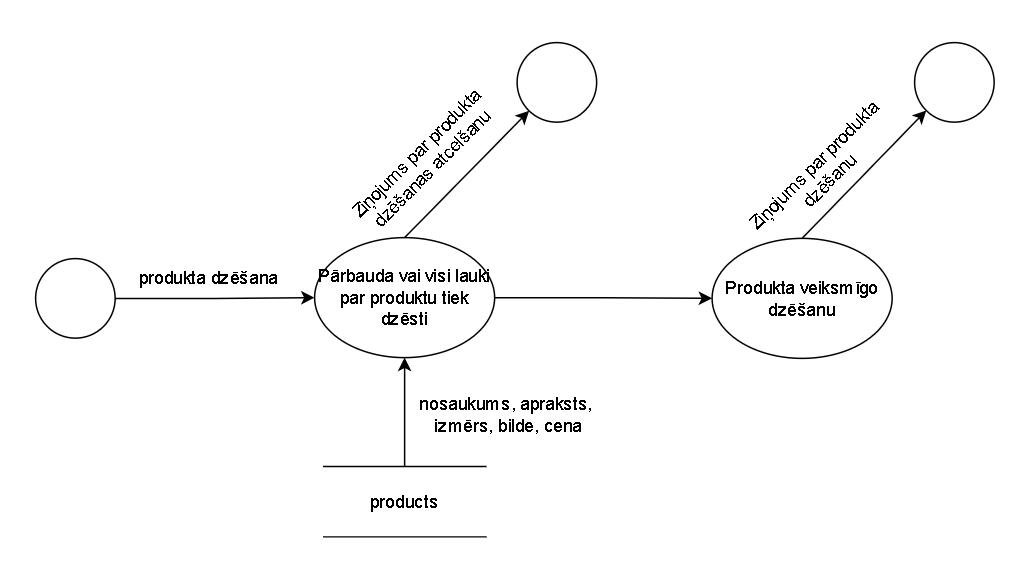
1. **produktu klāsta rediģēšana**: administrators var veikt produktu klāsta rediģēšanu. Administrators var pievienot produktus, rediģēt par tiem informāciju un dzēst to. Gadījumā, ja dotais produkts eksistē, par to tiks paziņots administratoram, un produkts netiks izveidots.

****

4.8. Datu plūsmas diagramma - produkta pievienošana

****

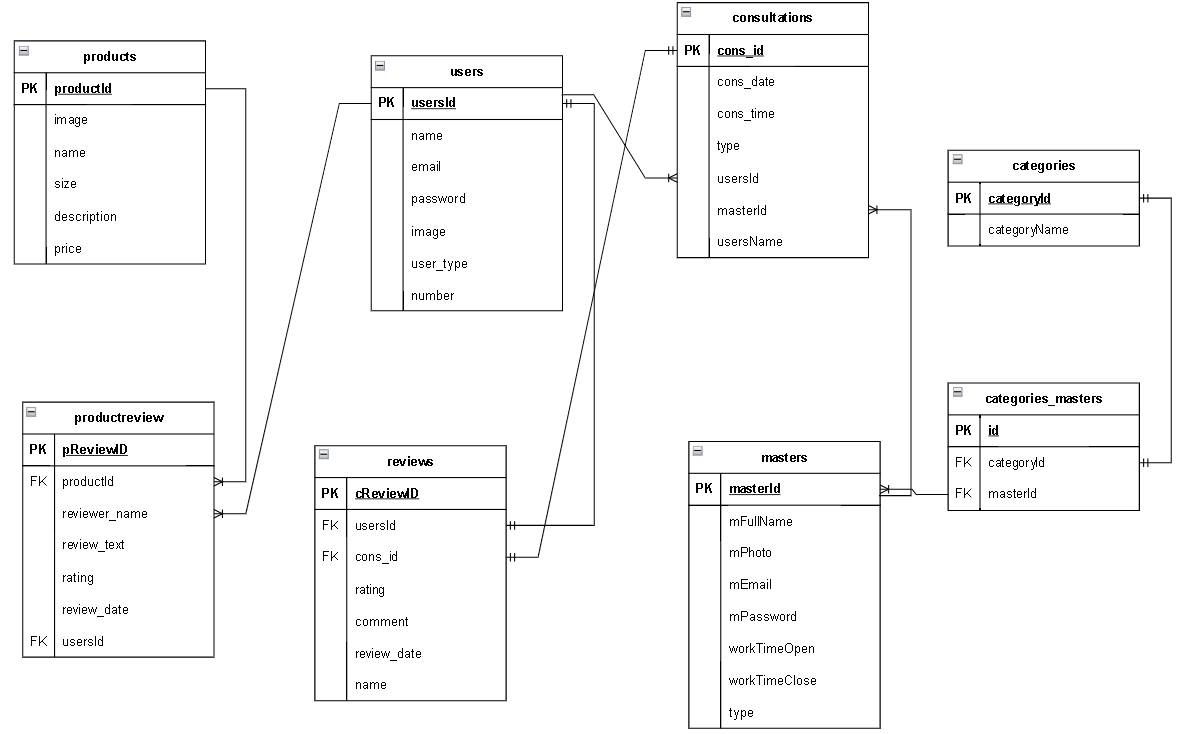
4.9. Datu plūsmas diagramma - produkta rediģēšana



4.10. Datu plūsmas diagramma - produkta dzēšana

# Datu struktūru apraksts

Datu bāze sastāv no 8 tabulām, kurās ir informācija par lietotājiem, konsultācijam, produktiem, meistariem un meistaru kategorijām.

****

5.1. att. Datu bāzes tabulu shēma

## 5.1. Datu bāzes fiziskā struktūra

Lietotāju tabula satur informāciju par lietotāja vārdu, epastu, paroli, bildi, lietotāja tipu, telefona numuru.

Tabula produkti satur informāciju par produktu bildi, nosaukumu, izmēru, aprakstu un cenu.

Tabula konsultācijas satur informāciju par konsultācijas datumu, laiku, tipu, primāro lietotāja un meistara atslēgu.

Tabula meistari satur informāciju par meistara vārdu un uzvārdu, meistara bildi, epastu un paroli.

Tabula produktu atsauksmes satur informāciju par produkta primāro atslēgu, lietotāja vārdu, kurš sniedza atsauksmi, tekstu, reitingu, datumu un lietotāja primāro atslēgu.

Tabula konsultāciju atsauksmes satur informāciju par lietotāja un konsultācijas primāram atslēgam, reitingu, komentāriju, datumu un lietotāja vārdu.

Tabula kategorijas satur informāciju par kategorijas nosaukumu.

Tabula kategorijas\_meistari ir starptabula, kas saista divas noteiktas tabulas: meistari un kategorijas. Tā satur informāciju par kategorijas primāro atslēgu un meistara primāro atslēgu.

Attiecības starp dotām tabulām:

* **“Lietotāji” - “Produktu atsauksmes”.** Attiecības viens pret daudziem. Lietotājs var sniegt vairākas atsauksmes par produktu, bet katram produktam tiek iesniegta konkrēta atsauksme no lietotāja.
* **“Lietotāji” – “Konsultācijas”.** Attiecības viens pret daudziem. Vienam lietotājam var būt vairākas konsultācijas, bet katra konsultācija pieder tikai vienam lietotājam.
* **“Meistari” - “Konsultācijas”. A**ttiecības viens pret daudziem. Katra konsultācija var tikt piešķirta vienam meistaram, bet katram meistaram var būt daudz konsultāciju.
* **“Produkts” - “Produkta atsauksmes”.** Attiecības viens pret daudziem. Katram produktam var būt vairākas atsauksmes, bet noteikta atsauksme var būt tikai vienam produktam.
* **“Konsultācija” - “Konsultācijas atsauksmes”.** Attiecības viens pret vienu. Katrai konsultācijai var būt viena atsauksme, un noteikta atsauksme pieder tikai vienai konsultācijai.
* **“Lietotājs” - “Konsultācijas atsauksmes”.** Attiecības viens pret vienu. Vienam lietotājam var būt viena atsauksme par konsultāciju.
* **“Kategorija” - “Kategorija\_Meistari”.** Attiecības viens pret vienu. Viena kategorija pieder vienam meistaram, kā arī viens meistars pieder pie konkrētas kategorijas.
* **“Meistars” - “Kategorija\_Meistari”.** Attiecības daudzi pret vienu. Vairāki meistari var piederēt vienai kategorijai, bet kategorija meistaram var būt tikai viena.

5.1.1. tabula

Tabulas **„users”** struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīme** |
| 1. | usersId | int | - | Lietotāja kārtas numurs |
| 2. | name | varchar | 100 | Lietotāja vārds |
| 3. | email | varchar | 100 | Lietotāja e-pasts |
| 4. | password | varchar | 150 | Lietotāja parole |
| 5. | image | varchar | 255 | Lietotāja bilde |
| 6. | user\_type | int | - | Lietotāja tips |
| 7. | number | varchar | 12 | Lietotāja numurs |

Tabula **“users”** satur primāro atslēgu **“usersId”**, kā arī satur lauku **“user\_type”**, kas nosaka, vai tas ir vienkāršs lietotājs, vai administrators, vai meistars.

5.1.2. tabula

Tabulas **„consultations”** struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīme** |
| 1. | cons\_id | int | - | Konsultācijas kārtas numurs |
| 2. | cons\_date | date | - | Konsultācijas datums |
| 3. | cons\_time | time | - | Konsultācijas laiks |
| 4. | type | varchar | 255 | Konsultācijas tips |
| 5. | usersId | int | - | Lietotāja kārtas numurs |
| 6. | masterId | int | - | Meistara kārtas numurs |
| 7. | userName | varchar | 255 | Lietotāja vārds, uzvārds |

Tabula “**consultations**” satur lietotāja kārtas numuru **“usersId”** un meistara kārtas numuru **“masterId”** (tas ir arējas atslēgas), tāpēc tabula ir saistīta ar tabulu “**users**” un tabulu “**masters**”.

5.1.3. tabula

Tabulas **„products”** struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīme** |
| 1. | productID | int | - | Produkta kārtas numurs |
| 2. | image | varchar | 255 | Produkta bilde |
| 3. | name | varchar | 255 | Produkta nosaukums |
| 4. | size | varchar | 50 | Produkta izmērs |
| 5. | description | varchar | 300 | Produkta apraksts |
| 6. | price | decimal | - | Produkta cena |

Tabula **“products”** satur primāro atslēgu **“productId”**.

5.1.4. tabula

Tabulas **“masters”** struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīme** |
| 1. | masterId | int | - | Meistara kārtas numurs |
| 2. | mFullName | varchar | 75 | Meistara vārds, uzvārds |
| 3. | mPhoto | varchar | 255 | Meistara bilde |
| 4. | mEmail | varchar | 255 | Meistara epasts |
| 5. | mPassword | varchar | 255 | Meistara parole |
| 6. | workTimeOpen | time | - | Darba laika sākums |
| 7. | workTimeClose | time | - | Darba laika beigas |
| 8. | type | int | 2 | Lietotaja tips kas definets ar 2 |

Tabula **“masters”** satur meistara primāro atslēgu **“masterId”**.

5.1.5. tabula

Tabulas **“productreview”** struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīme** |
| 1. | pReviewID | int | - | Produkta atsauksmes kārtas numurs |
| 2. | productId | varchar | 255 | Produkta kārtas numurs |
| 3. | reviewer\_name | varchar | 255 | Atsauksmes izveidotaja vārds, uzvārds |
| 4. | review\_text | varchar | 50 | Atsauksmes teksts |
| 5. | rating | int | - | Atsauksmes vērtējums |
| 6. | review\_date | date | - | Atsauksmes datums |
| 7. | usersId | Int | - | Lietotāja kārtas numurs |

Tabula “**productreview**” satur primāro atslēgu **“pReviewID”**, kā arī satur divas arējas atslēgas: produkta kārtas numuru **“productId”** un lietotāja kārtas numuru **“usersId”**. Tabula ir saistīta ar tabulām **“products”** un **“users”**.

5.1.6. tabula

Tabulas **“reviews”** struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīme** |
| 1. | cReviewID | int | - | Konsultācijas atsauksmes kārtas numurs |
| 2. | usersId | varchar | 255 | Lietotāja kārtas numurs |
| 3. | cons\_id |  |  | Konsultācijas kārtas numurs |
| 4. | reviewer\_name | varchar | 255 | Atsauksmes izveidotaja vārds, uzvārds |
| 5. | review\_text | varchar | 50 | Atsauksmes teksts |
| 6. | rating | int | - | Atsauksmes vērtējums |
| 7. | review\_date | date | - | Atsauksmes datums |

Tabula “**reviews**” satur lietotāja kārtas numuru **“usersId”** un konsultācijas kārtas numuru **“cons\_id”**, tāpēc tabula ir saistīta ar tabulu “**Lietotāji”** un tabulu “**Konsultācijas**”**.**

5.1.7. tabula

Tabulas **“categories\_masters”** struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīme** |
| 1. | id | int | - | Kārtas numurs lai saistītu meistaru un kategoriju |
| 2. | masterId | int | - | Meistara kārtas numurs |
| 3. | categoryId | int | - | Kategorijas kārtas numurs |

Tabula **“categories\_masters”** satur tikai trīs atslēgas: primāra atslēga **“id”**, lai saistītu meistaru un kategoriju, jo šī tabula ir starptabula, meistara arējo atslēgu **“masterId”** un kategorijas arējo atslēgu **“categoryId”**. Šī tabula ir saistīta ar tabulām **“masters”** un **“categories”**.

5.1.8. tabula

Tabulas **“categories”** struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīme** |
| 1. | categoryId | int | - | Kategorijas kārtas numurs |
| 2. | categoryName | varchar | 255 | Kategorijas nosaukums |

Tabula **“categories”** satur primāro atslēgu **“categoryId”**, kas tiek izmantots tabulā **“categories\_masters”**, lai meistaram varētu piešķirt kategoriju.

# Lietotāja ceļvedis

## 6.1. Sistēmas prasības aparatūrai un programmatūrai

#### Lai veiksmīgi palaistu un darbinātu sistēmu, ir nepieciešams nodrošināt atbilstošu aparatūru un programmatūru, kas atbilst šeit norādītajām prasībām. Tādēļ lūdzu, iepazīstieties ar zemāk sniegtajām sistēmas prasībām, kas ietver gan aparatūras, gan programmatūras prasības:

#### 1. Aparatūras prasības

##### 1.1. Serveris:

###### Minimālās prasības:

* Procesors: 4 kodolu CPU, 2.5 GHz
* Atmiņa: 8 GB RAM
* Diska vieta: 100 GB SSD
* Tīkls: Gigabitu Ethernet

###### Ieteicamās prasības:

* Procesors: 8 kodolu CPU, 3.0 GHz vai augstāks
* Atmiņa: 16 GB RAM vai vairāk
* Diska vieta: 500 GB SSD vai vairāk
* Tīkls: 10 Gigabitu Ethernet

##### 1.2. Darbstacijas (lietotājiem):

###### Minimālās prasības:

* Procesors: 2 kodolu CPU, 1.8 GHz
* Atmiņa: 4 GB RAM
* Diska vieta: 50 GB HDD
* Ekrāna izšķirtspēja: 1366x768
* Interneta pieslēgums: stabils interneta pieslēgums ar 5 Mbps lejupielādes ātrumu

###### Ieteicamās prasības:

* Procesors: 4 kodolu CPU, 2.5 GHz vai augstāks
* Atmiņa: 8 GB RAM vai vairāk
* Diska vieta: 128 GB SSD vai vairāk
* Ekrāna izšķirtspēja: 1920x1080
* Interneta pieslēgums: stabils interneta pieslēgums ar 20 Mbps lejupielādes ātrumu vai ātrāks

#### 2. Programmatūras prasības

##### 2.1. Serveris:

* Operētājsistēma: Linux (piemēram, Ubuntu Server 20.04 LTS) vai Windows Server 2019
* Web serveris: Apache 2.4 vai jaunāks
* Datubāze: MySQL 8.0 vai jaunāks
* Programmatūras ietvars: PHP 8.0 vai jaunāks
* Papildus: OpenSSL, cURL, PHPMyAdmin

##### 2.2. Darbstacijas (lietotājiem):

* Operētājsistēma: Windows 10 vai jaunāka, macOS Catalina vai jaunāka
* Pārlūkprogramma: Google Chrome 89 vai jaunāka, Microsoft Edge 89 vai jaunāka, Mozilla Firefox 87 vai jaunāka
* Papildus programmatūra: Adobe Acrobat Reader, Java Runtime Environment (JRE) 8 vai jaunāka

##### 2.3. Izstrādes vide:

* Kods redaktors/IDE: Visual Studio Code v.1.85 ar PHP un JavaScript atbalstu
* Lokālais serveris: XAMPP v.8.2.12, kas ietver Apache, MySQL un PHP
* Versiju kontrole: Git (ar GitHub vai GitLab kā attālinātu repozitoriju)

#### 3. Nefunkcionālās prasības

* Uzticamība: 99.9% pieejamība
* Drošība: Datu drošība ar SSL šifrēšanu, regulāriem drošības atjauninājumiem un stingriem piekļuves kontroles mehānismiem
* Veiktspēja: Reakcijas laiks ne ilgāk kā 2 sekundes normālas slodzes apstākļos
* Mērogojamība: Viegla mērogojamība, lai apstrādātu pieaugošu lietotāju un datu apjomu
* Lietojamība: Intuitīvas un viegli lietojamas lietotāja saskarnes, atbalstot dažādas ierīces un ekrānu izmērus

## 6.2. Sistēmas instalācija un palaišana

Lai veiktu sistēmas instalāciju un palaišanu, ir jāievēro vairāki soļi, sākot no aparatūras sagatavošanas līdz programmatūras uzstādīšanai un konfigurēšanai.

#### 6.2.1. Aparatūras sagatavošana

Pirms sākat instalāciju, ir svarīgi nodrošināt, ka visi nepieciešamie datori un serveri atbilst minimālajām un ieteicamajām aparatūras prasībām.

1. **Serveris:**
   1. Pārliecinieties, vai serverim ir atbilstošs procesors, atmiņa, diskvieta un tīkla savienojums, kā norādīts minimālajās un ieteicamajās prasībās.
   2. Uzstādiet operētājsistēmu, piemēram, Ubuntu Server 20.04 LTS vai Windows Server 2019, atkarībā no jūsu izvēles un vajadzībām.
2. **Darbstacijas (lietotājiem):**
   1. Pārliecinieties, ka visām darbstacijām ir nepieciešamais procesors, atmiņa, diskvieta, ekrāna izšķirtspēja un interneta pieslēgums, saskaņā ar minimālajām un ieteicamajām prasībām.
   2. Uzstādiet operētājsistēmu, piemēram, Windows 10 vai macOS Catalina, lai nodrošinātu saderību ar citām programmatūras sastāvdaļām.

#### 6.2.2. Programmatūras instalācija un konfigurācija

Kad aparatūra ir gatava, nākamais solis ir programmatūras uzstādīšana un konfigurēšana uz servera un darbstacijās:

1. **Serveris:**
   1. Lejupielādējiet un instalējiet Apache web serveri, MySQL datubāzi un PHP programmēšanas valodu, izmantojot atbilstošus avotus vai oficiālās mājaslapas.
   2. Konfigurējiet katru komponentu, iestatot tos saskaņā ar jūsu sistēmas prasībām un drošības standartiem.
   3. Nodrošiniet, ka SSL šifrēšana ir iestatīta, lai aizsargātu datu pārsūtījumus starp serveri un klientiem.

**2. Darbstacijas (lietotājiem):**

* 1. Instalējiet nepieciešamās pārlūkprogrammas un papildus programmatūru, piemēram, Adobe Acrobat Reader un Java Runtime Environment (JRE), lai nodrošinātu saderību un pilnvērtīgu lietošanas pieredzi.
  2. Konfigurējiet darbstacijas, lai tās būtu gatavas sazināties ar serveri un izmantot sistēmu, piemēram, konfigurējot interneta pieslēgumu un pārlūkprogrammas iestatījumus.

#### 6.2.3. Sistēmas palaišana un testēšana

Kad visa programmatūra ir uzstādīta un konfigurēta, varat turpināt ar sistēmas palaišanu un pirmo testēšanu:

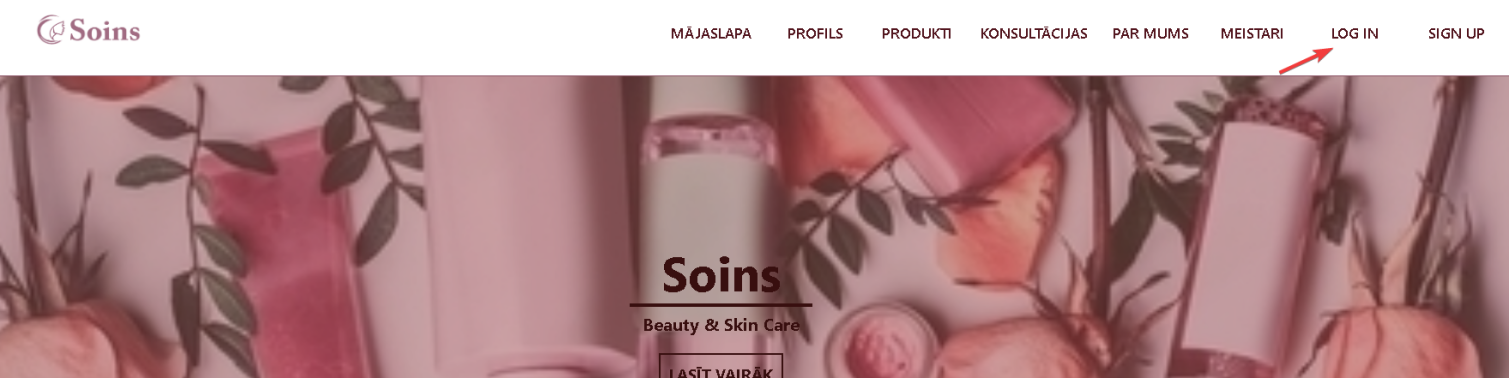
1. Palaistiet serveri un pārliecinieties, ka visi nepieciešamie pakalpojumi (web serveris, datubāze) ir startējušies un darbojas pareizi.
2. Pārbaudiet, vai darbstacijas var veiksmīgi piekļūt serverim un sistēmai, izmantojot tās piedāvātās funkcijas.
3. Veiciet sākotnējo sistēmas konfigurāciju, piemēram, izveidojot datubāzi un lietotāja kontus, un pārbaudiet, vai tie darbojas kā paredzēts.
4. Veiciet drošības pārbaudes un nodrošiniet, ka visas drošības iestatījumi un atjauninājumi ir uzstādīti un darbojas kā paredzēts.
5. Testējiet sistēmas veiktspēju, izmantojot to ar dažādiem slodzēm, lai pārliecinātos, ka tā darbojas efektīvi un reaģē uz lietotāju pieprasījumiem.

Pēc šiem soļiem jūsu skaistumkopšanas uzskaites sistēma būs gatava darbam, nodrošinot augstu klientu apmierinātības līmeni un uzlabojot uzņēmējdarbības efektivitāti.

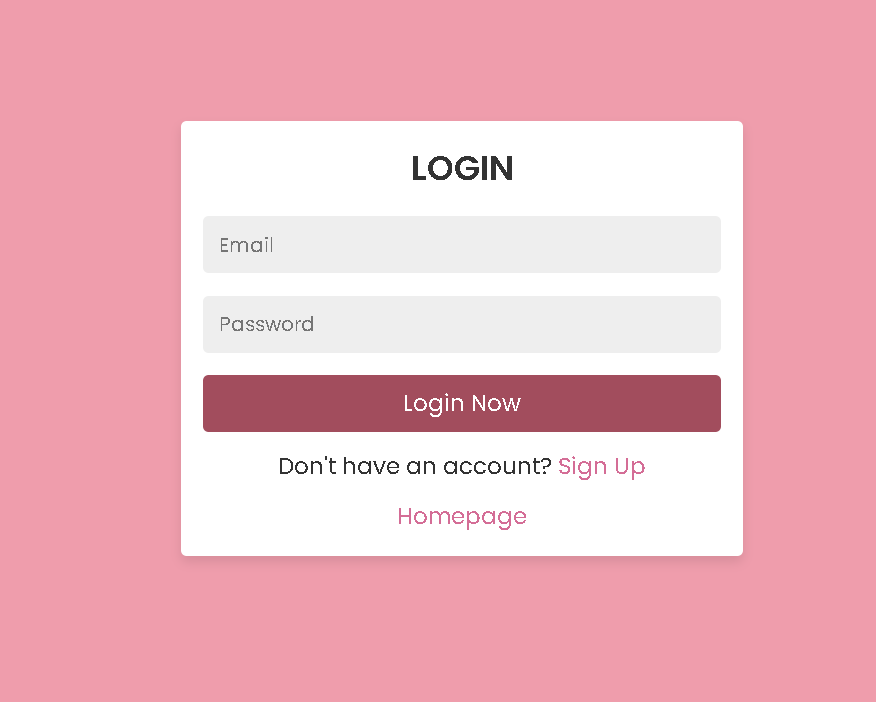
## 6.3. Programmas apraksts

1. **Lietotāju reģistrācija un autorizēšanas**

Sākumā mums ir jāatver mājaslapa: <http://localhost/index.php,> tad pārējam uz lapu “Log In” un ienācam savā profilā. (skat. 6.1. un 6.2. att.)



6.1. att. Mājaslapa.



6.2. att. Log In (Ieejas) lapa

Gadījumā, ja neeksistē profils, spiežam uz pogu “Sign Up”. Tā ir gan mājaslapas panelī, gan Log In lapā. Ievadam nepieciešamo informāciju: vārdu, epastu, numuru, paroli un vēlreiz paroli (to ir jāievada divas reizes, lai būtu pārliecība, ka esat ievadījuši to pareizi). Obligāti ir jāaizpilda visi lauki, savukārt, ja nav aizpildīts kaut viens lauks, netiksiet reģistrēti. Kad esat aizpildījuši visus laukus, spiežam pogu “Register”. (skat. 6.3. att.)

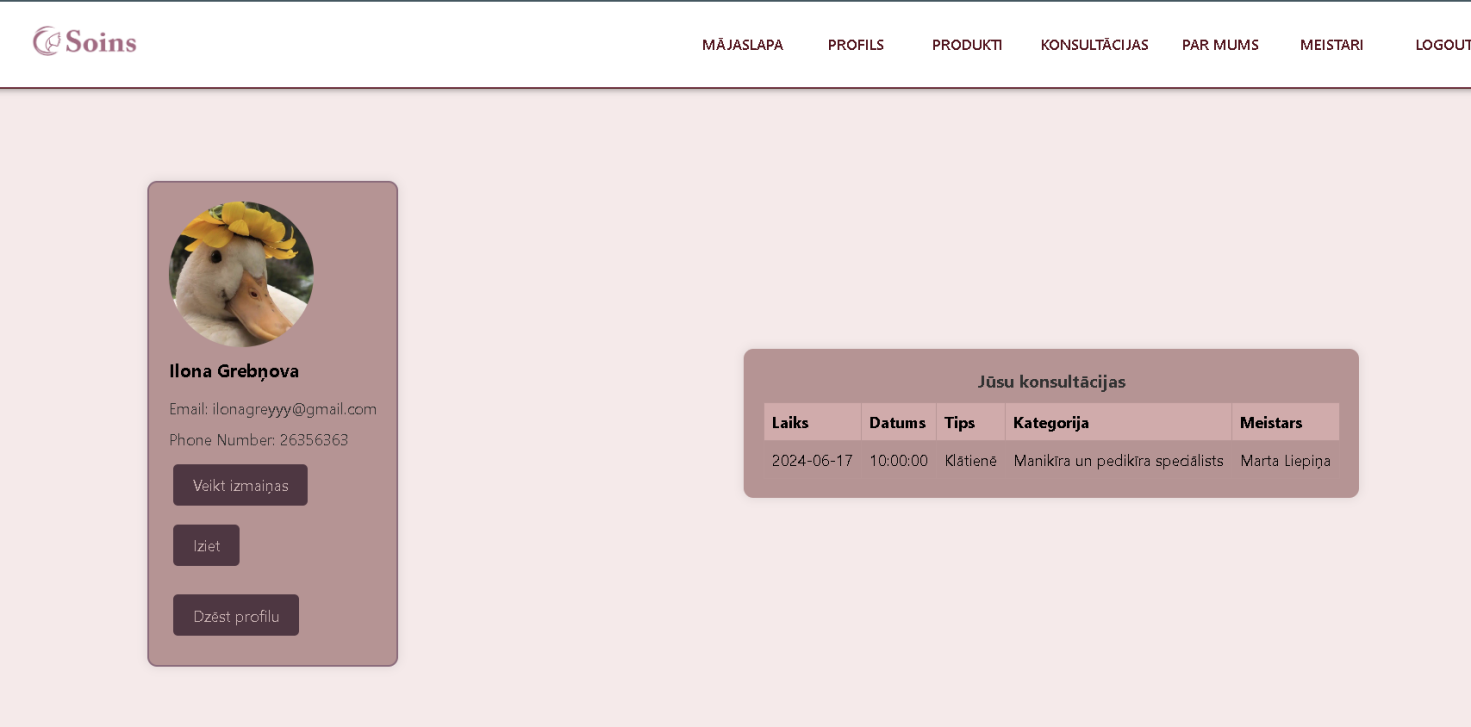


6.3. att. Sign Up (Reģistrēšanas) lapa

Pēc veiksmīgas reģistrēšanas lietotāju pārved uz citu lapu, kur viņam parādas ziņa par veiksmīgo reģistrāciju (skat. 6.4. att.), un lietotājs var izvēlēties, pāriet uz ieejas lapu vai uz mājaslapu.

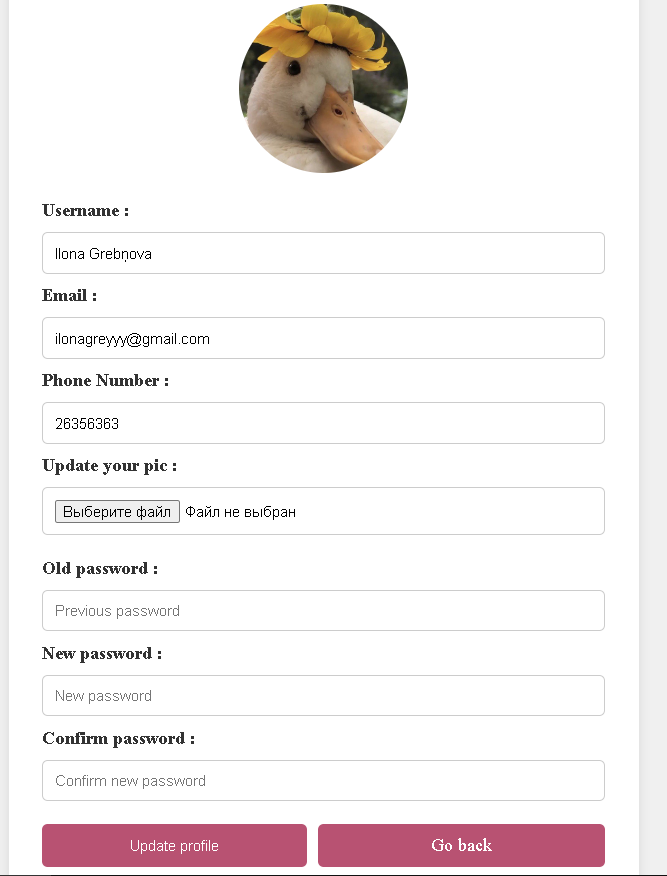


6.4. att. Veiksmīgas reģistrācijas ziņojums

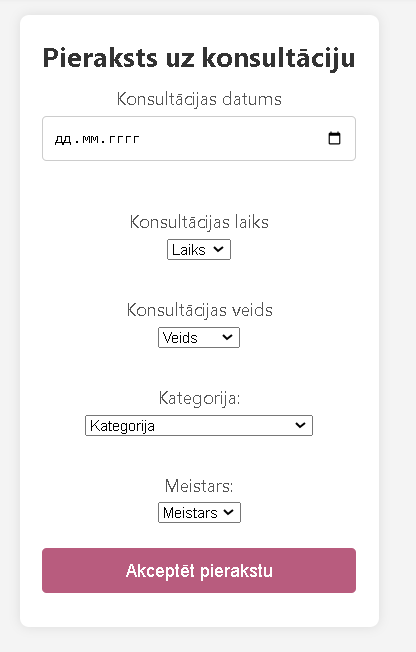


6.5. att. Lietotāja profils

Navigācijas panelī uzklikšķinot uz pogu “Profils”, lietotājs pāries uz savu profilu, kur radīsies informācija par lietotāju, un par viņa pieteiktām konsultācijam (pierakstu uz konsutāciju lietotājs veic lapā konsultācijas, kur ir speciāla aizpildes forma, skat. 6.7. att.). Ja lietotājs vēlas veikt izmaiņas, piemēram, nomainīt epastu vai telefona numuru, ir jāuzklikšķina poga “Veikt izmaiņas”, lietotāju novedīs uz lapu, kur var veikt izmaiņas (skat. 6.6. att.).

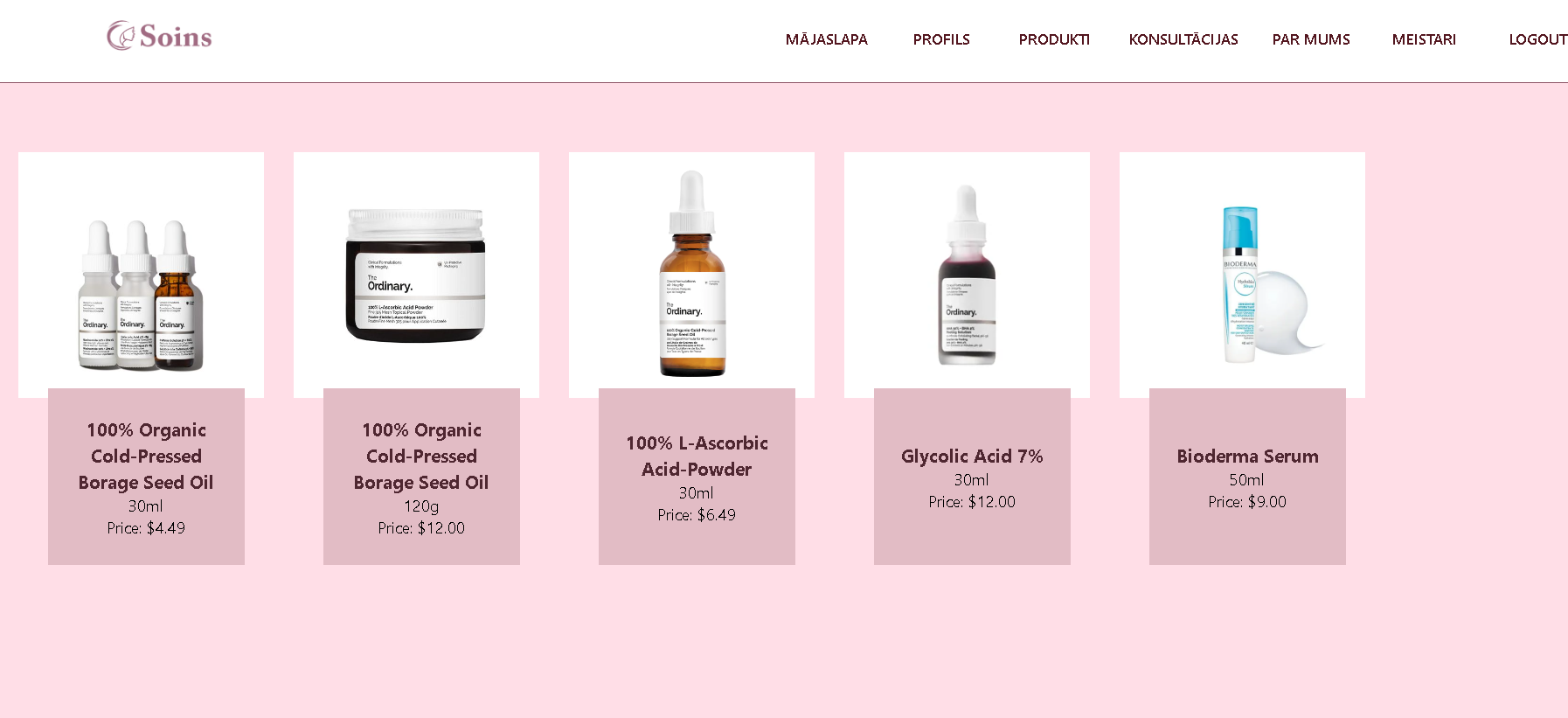


6.6. att. Lietotāja datu izmainīšana

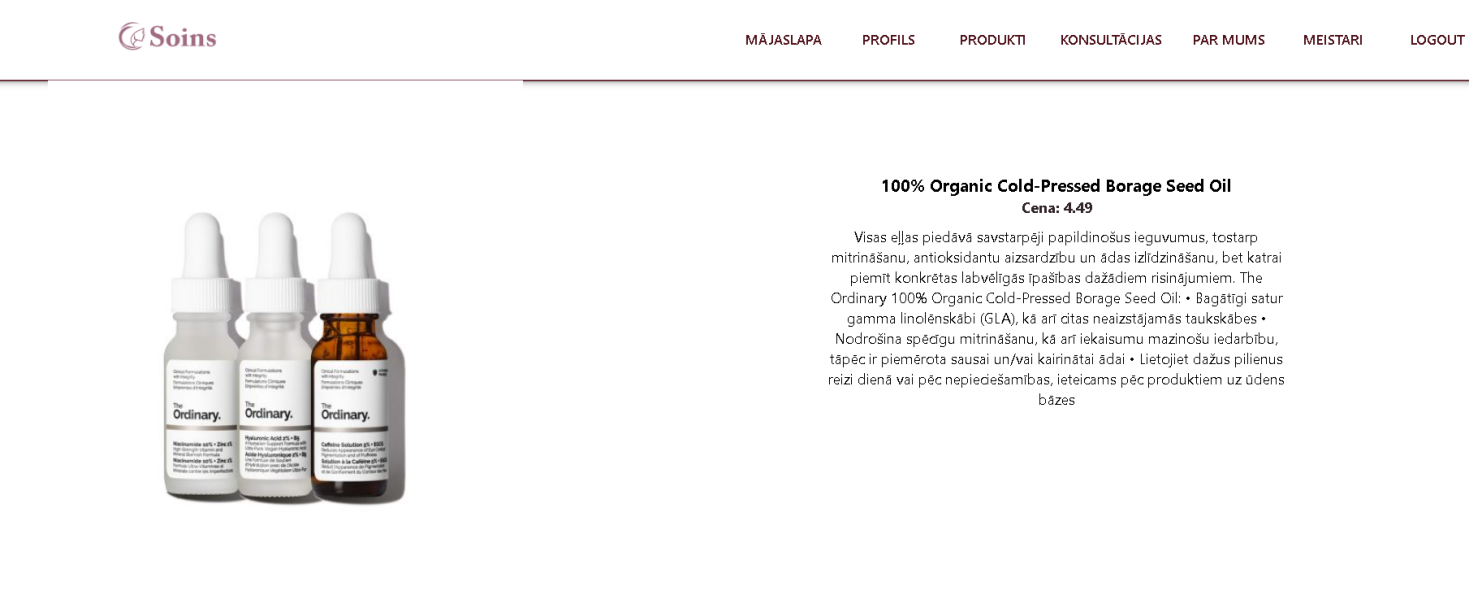


6.7. att. Forma konsultācijas lapā, lai pierakstītos uz konsultāciju

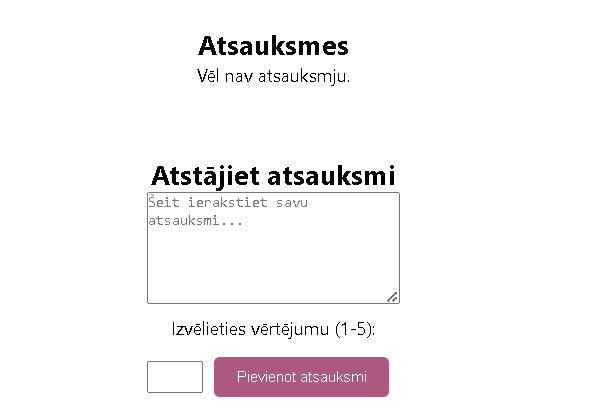
Lietotājs var apskatīt produktus, kas ir pieejami salonā, pagaidām viņiem nav interneta-veikals, produktus var iegādāties atnācot pie viņiem salonā. Uzspiežot uz kādu no produkta nosaukumiem, var atvērt lapu ar detalizēto produkta aprakstu, un atstāt savu atsauksmi par produktu. (skat. 6.8.att, 6.9.att. un 6.10. att.)



6.8. att. Produktu klāsts

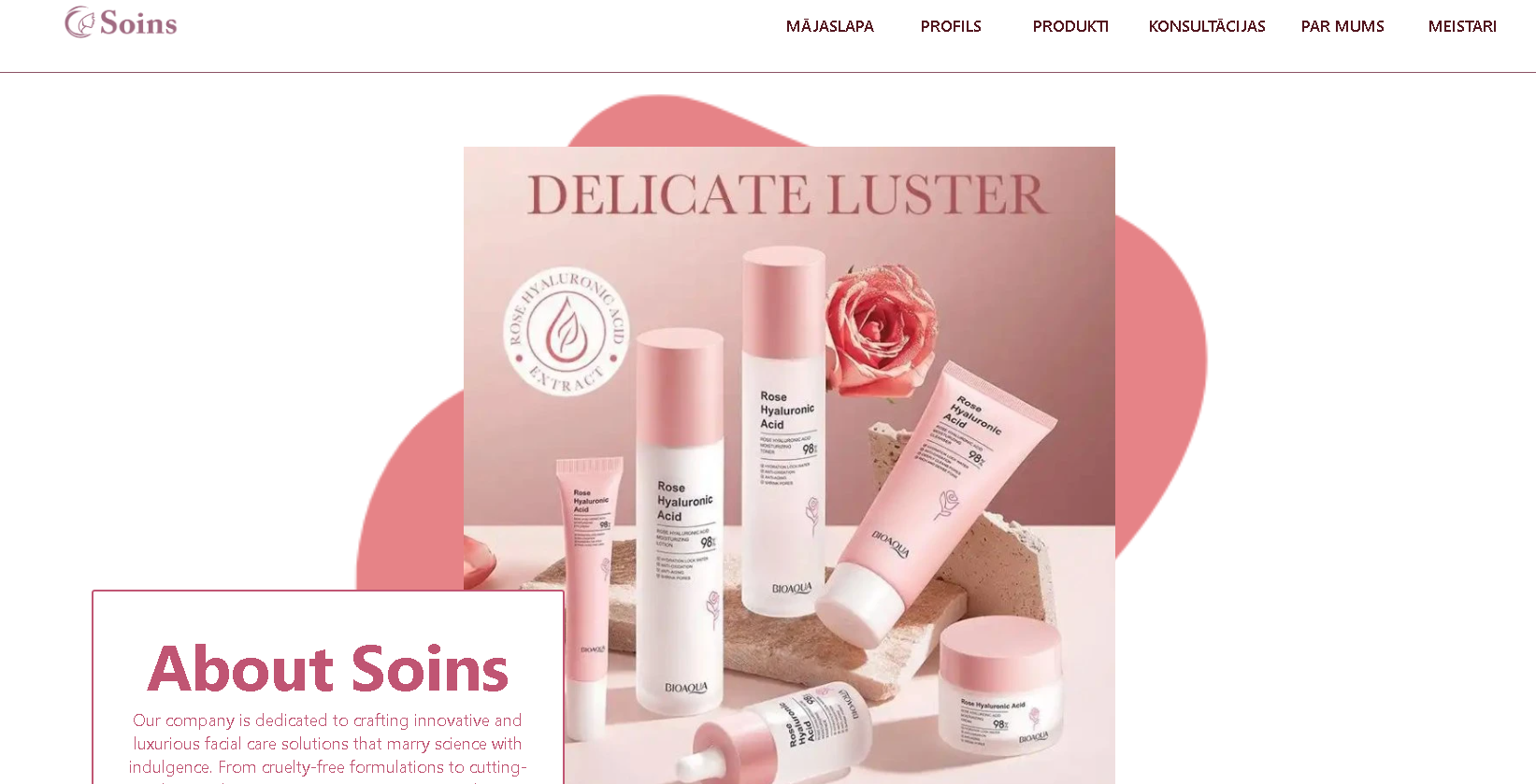


6.9. att. Produkta detalizētais apraksts

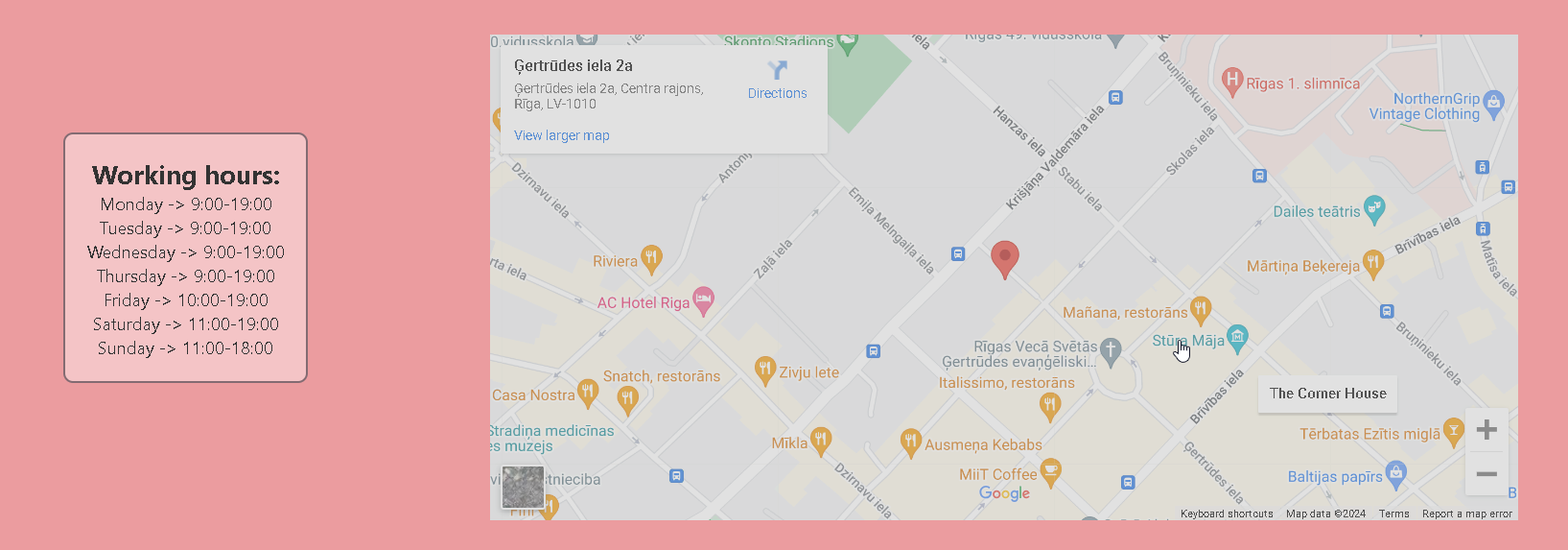


6.10. att. Atsauksmes forma un atsauksmes kas tiks parādītas

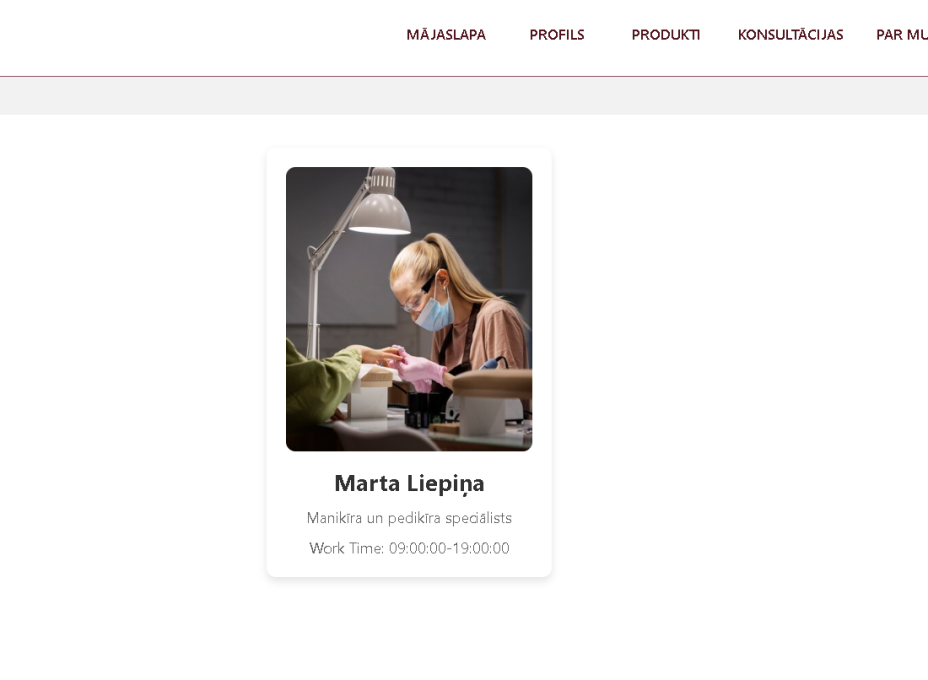
Lapā “Par mums”, lietotājs var apskatīt salona aprakstu, uzzināt darba laiku un atrašanas vietu (skat.6.11.att., 6.12.att.)



6.11. att. “Par mums” - apraksts par salonu



6.12. att. “Par mums” - darba laiks un atrašanas vieta

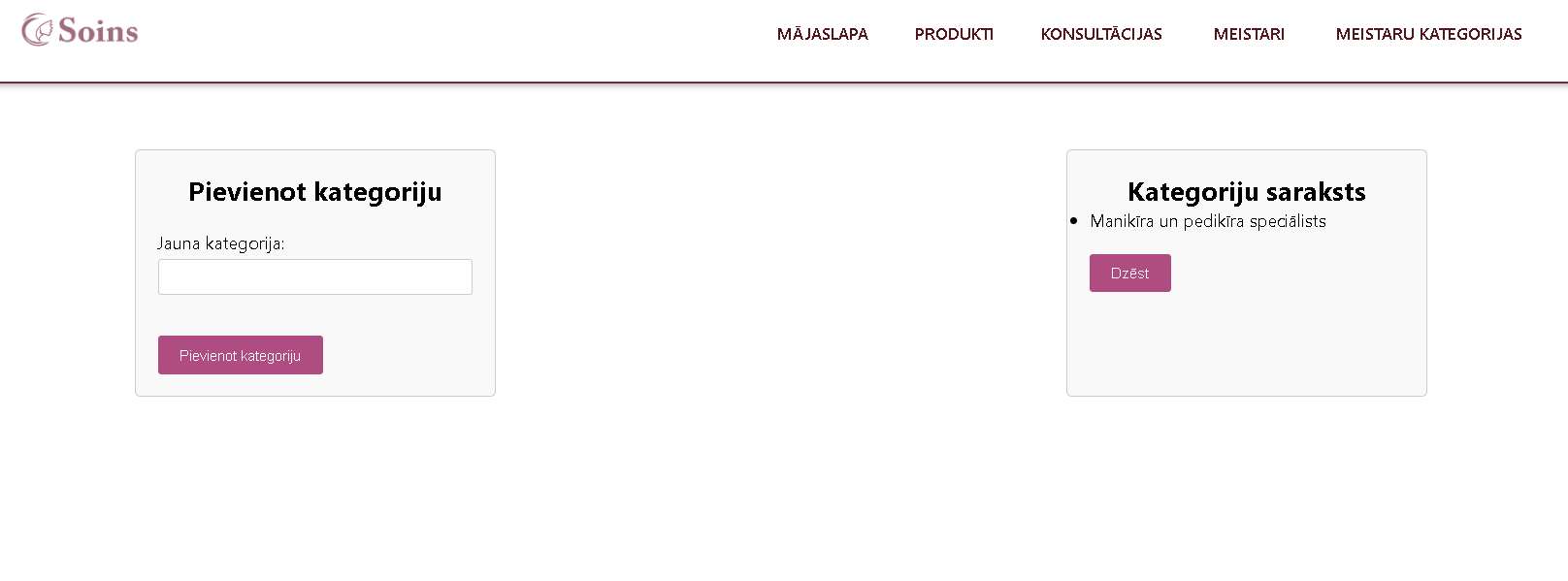


6.13. att. “Meistari” lapa

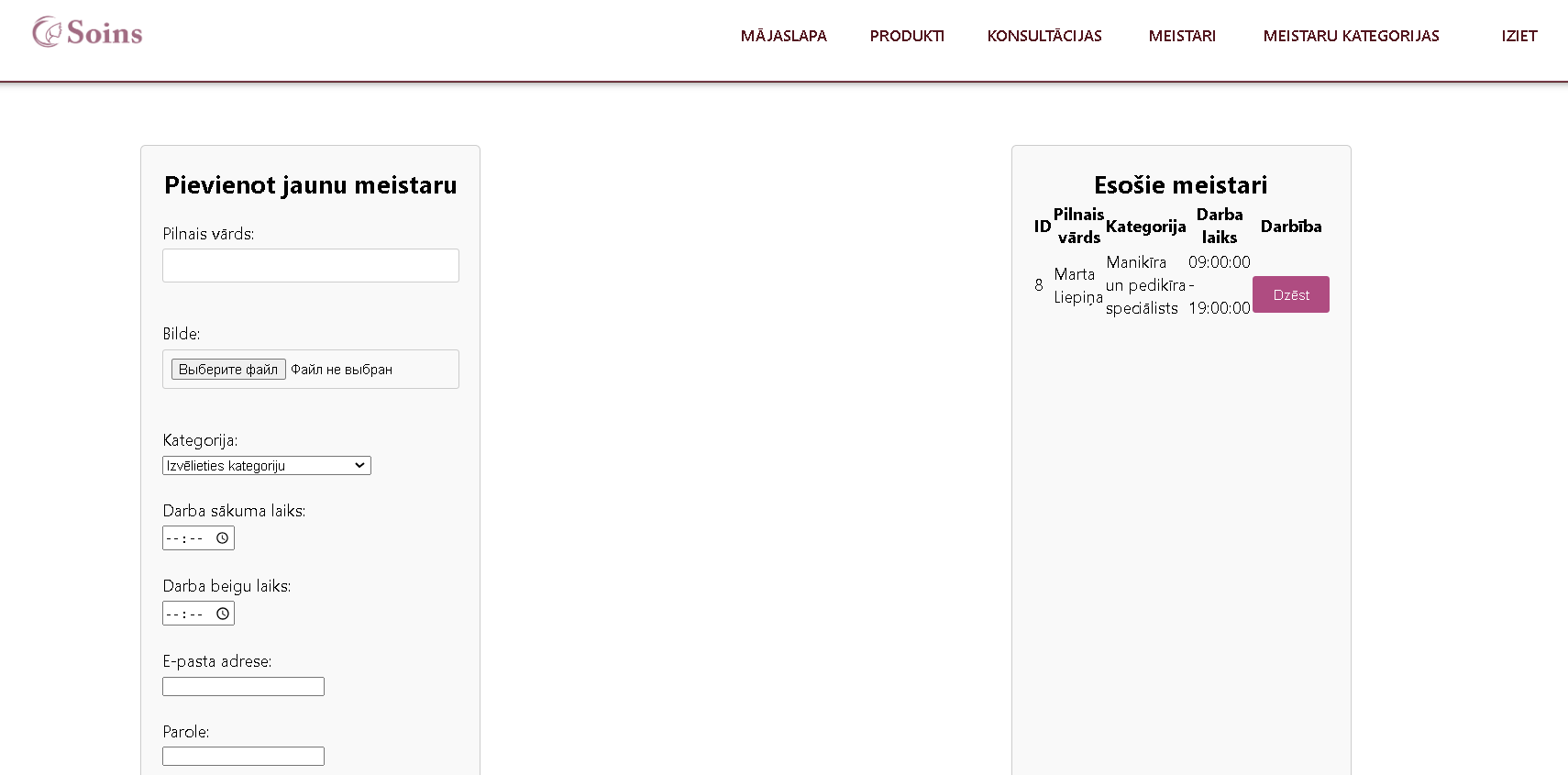
Administratora panelis, kā arī lapas, atšķiras no lietotāja paneļa un pieejamam lapām (skat. 6.14.att., 6.15.att., 6.16.att., 6.17.att. utt.,)



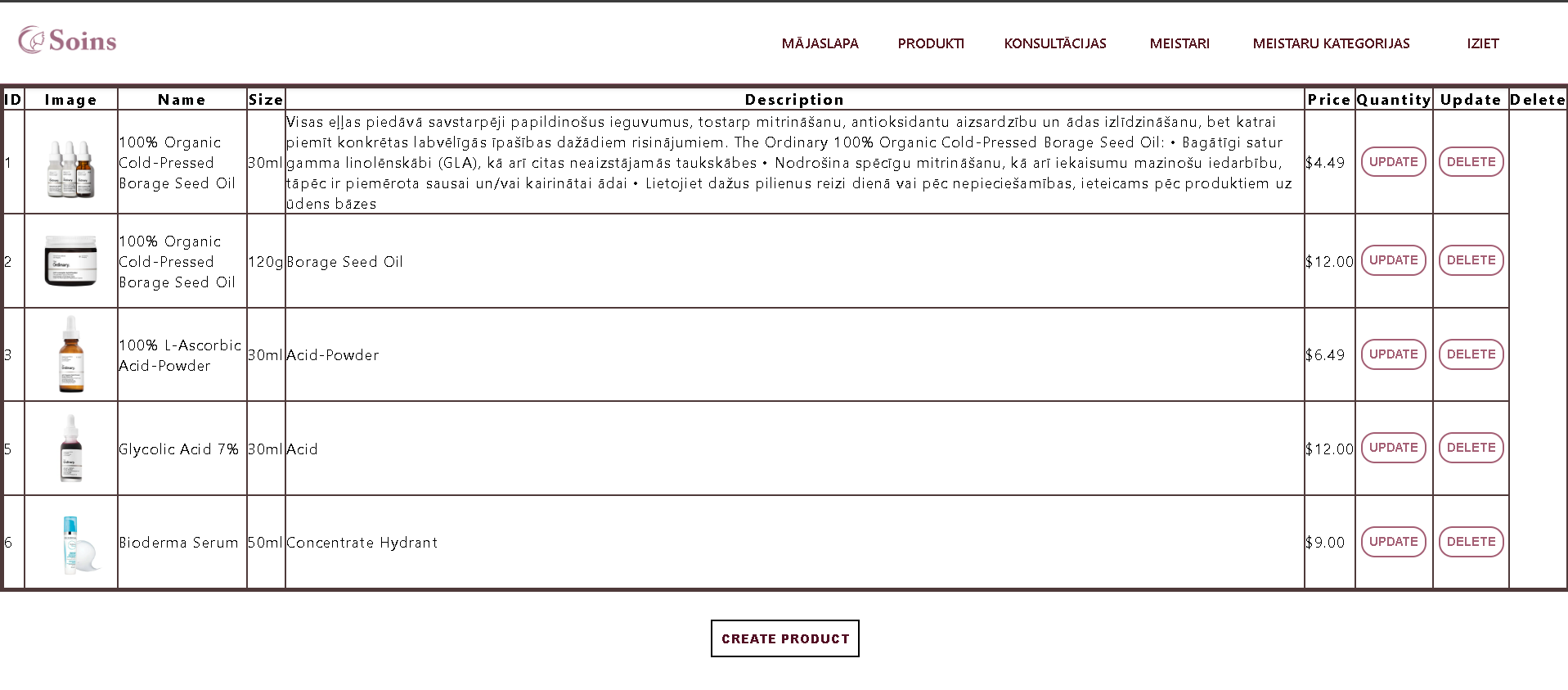
6.14. att. Administratora panelis



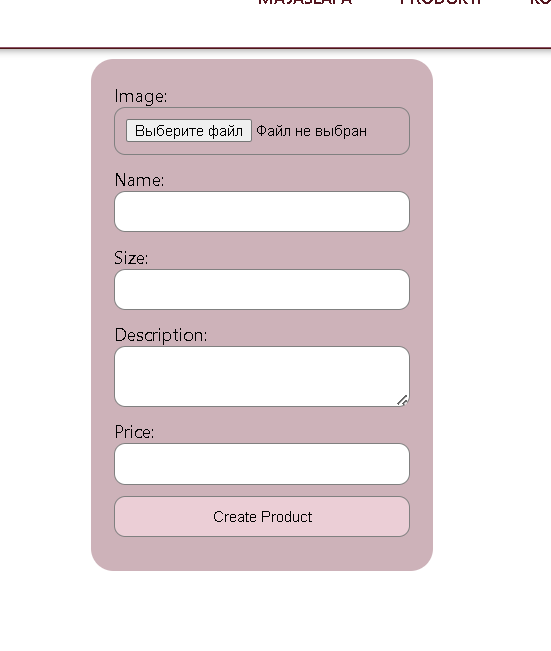
6.15. att. Meistaru kategorijas lapa (administrators)



6.16. att. Meistaru pievienošanas lapa (administrators)



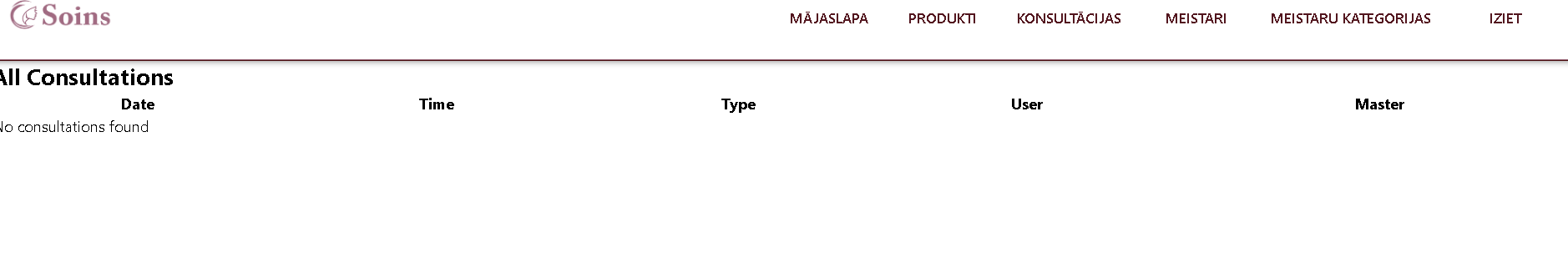
6.17. att. Produktu pievienošanas lapa (administrators)



6.18. att. Produktu pievienošanas forma (administrators)



6.19. att. Produkta informācijas atjaunināšanas forma (administrators)



6.20. att. Pievienoto lietotāju konsultāciju apskate (administrators)

## 6.4. Testa piemērs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Prasības numurs | Prasības nosaukums | Ievaddati/situācijas apraksts | Sagaidāmais rezultāts | Statuss |
| 1. | 1. | Jauna lietotāja reģistrācija | Pareizi dati  lietotāja vārds, lietotāja e-pasts,  1. parole, 2. parole | Datu bāzē tiek pievienots ieraksts, saskarnē parādās paziņojums par veiksmīgu reģistrāciju | Pareizi |
| 2. | Lietotāja vārds = MarijaSS | Paziņojums par jau eksistējošo lietotāju | Pareizi |
| 3. | Lietotāja e-pasts = marija!$gmail.com | Paziņojums par e-pasta nepareizo formātu | Pareizi |
| 4. |  | Produkta pievienošana | Peles klikšķis uz pogas “Pievienot” | Tiek pievienots jauns produkts kas parādās preču klāsta (lietotāju pusē) un tabulā ar produktiem (administratora pusē) | Pareizi |
| 5. | Produkta bilde | Pievienotai produkta bildei jārādās gan administratorā tabulā, gan produktu klāstā lietotājiem | Pareizi |
| 6. |  | Konsultācijas pievienošana | Peles klikšķis uz pogas “Pievienot” | Lietotājs izvēlās konkrētu datumu un laiku lai veiktu reģistrāciju uz konsultāciju | Pareizi |
| 7. | Pievienošanas datums un laiks | Lietotājam rādās tikai tas laiks, kurā nav pieteikta konsultācija | Pareizi |

Visas kļūmes, kas tika atrastas testējot kodu, tika izlabotas, lai programma priekš lietotājiem, meistariem un administratoriem darbotos pareizi.

# Nobeigums

Pēc rūpīgas analīzes un detalizētas izstrādes laika posmā, kvalifikācijas darba izstrādes laikā mēs esam veiksmīgi izveidojuši sistēmas modeli un projektējuši to atbilstoši specifikācijās noteiktajām prasībām. Mēs esam īstenojuši šo procesu, izmantojot dažādas programmēšanas valodas: PHP, JavaScript, HTML un CSS, kā arī izmantojot Visual Studio Code un XAMPP kā attīstības rīkus.

Pašlaik šis modelis un projektēšanas process tiek izmantots, lai virzītos uz priekšu ar programmatūras izstrādi. Tas jau ir noderīgs, lai saprastu sistēmas struktūru un darbību, un tas tiks turpmāk izmantots kā pamats kodēšanai un testēšanai. Turklāt, balstoties uz šo pamatu, iespējams attīstīt sistēmu vēl tālāk, paplašinot funkcionalitāti vai pielāgojot to jaunām vajadzībām.

Nobeigumā, izmantojot šo struktūru un analīzi, mēs varam doties uz priekšu ar drošību un pārliecību, zinot, ka mūsu programmatūras izstrādes process ir rūpīgi izstrādāts un pamatots. Tas nodrošina iespēju ne tikai izveidot kvalitatīvu produktu tagad, bet arī attīstīt to nākotnē uz kaut ko lielāku un inovatīvāku.

# Informācijas avoti

* Diagrammu veidošana - <https://app.diagrams.net/> - (Resurss apskatīts 13.11.2023)
* Informācija par PHP valodu - <https://www.php.net/> - (Resurss apskatīts 13.11.2023)
* Skiču veidošanai - <https://www.figma.com/> - (Resurss apskatīts 13.11.2023)
* Informācija par programmatūras prasībām - <https://lv.itpedia.nl/2017/11/30/kenmerken-voor-software-requirements/> - (Resurss apskatīts 13.11.2023)
* Prasību specifikācija - [https://argondigital.com/blog/product-management/requirements-vs-specifications-create-a-shared-vocabulary/#:~:text=A%20requirement%20specification%20is%20a,and%20maintenance%20of%20the%20product.](https://argondigital.com/blog/product-management/requirements-vs-specifications-create-a-shared-vocabulary/" \l ":~:text=A requirement specification is a,and maintenance of the product.) - (Resurss apskatīts 13.11.2023)
* ER-diagramma - <https://creately.com/guides/er-diagrams-tutorial/> - (Resurss apskatīts 13.11.2023)
* Funkcionālas dekompozīcijas diagramma - <https://binaryterms.com/functional-decomposition.html> - (Resurss apskatīts 13.11.2023)
* Datu bāzes tabulu shēma - <https://www.educative.io/blog/what-are-database-schemas-examples> - (Resurss apskatīts 13.11.2023)
* Datubāzes pieslēgšana izmantojot PHP valodu - <https://www.simplilearn.com/tutorials/php-tutorial/php-with-sql> - (Resurss apskatīts 13.11.2023)

PIELIKUMI

1.pielikums. Programmas pirmteksts

**// Lietotāja profila lapa un stils**

<?php

include\_once 'blocks/header.php';

include 'includes/dbh.inc.php';

$user\_id = $\_SESSION['user\_id'];

if (!isset($user\_id)) {

header('location:login.php');

}

if (isset($\_GET['logout'])) {

unset($user\_id);

session\_destroy();

header('location:login.php');

}

// Query to fetch user's consultations

$consultations\_query = mysqli\_query($conn, "SELECT consultations.\*, categories.categoryName AS master\_category, masters.mFullName

FROM `consultations`

INNER JOIN `categories\_masters` ON consultations.masterId = categories\_masters.masterId

INNER JOIN `categories` ON categories\_masters.categoryId = categories.categoryId

INNER JOIN `masters` ON consultations.masterId = masters.masterId

WHERE consultations.usersId = '$user\_id'

ORDER BY consultations.cons\_date DESC, consultations.cons\_time DESC") or die('Query failed: ' . mysqli\_error($conn));

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Home</title>

<link rel="stylesheet" href="css/profile.css">

</head>

<body>

<div class="pcontainer">

<div class="profile">

<?php

$select = mysqli\_query($conn, "SELECT \* FROM `users` WHERE usersId = '$user\_id'") or die('query failed');

if (mysqli\_num\_rows($select) > 0) {

$fetch = mysqli\_fetch\_assoc($select);

}

if ($fetch['image'] == '') {

echo '<img src="images/default-avatar.png">';

} else {

echo '<img src="uploaded\_img/'.$fetch['image'].'">';

}

?>

<h3><?php echo $fetch['name']; ?></h3>

<p>Email: <?php echo $fetch['email']; ?></p>

<p>Phone Number: <?php echo $fetch['number']; ?></p>

<a href="update\_profile.php" class="btn">Veikt izmaiņas</a>

<a href="profile.php?logout=<?php echo $user\_id; ?>" class="btn">Iziet</a>

<a href="deleteuser.php?id=<?php echo $user\_id; ?>" class="delete-btn">Dzēst profilu</a>

<a href="download\_pdf.php" class="btn">Lejupielādēt PDF</a>

</div>

<div class="consultations-container">

<h2>Jūsu konsultācijas</h2>

<table>

<tr>

<th>Laiks</th>

<th>Datums</th>

<th>Tips</th>

<th>Kategorija</th>

<th>Meistars</th>

</tr>

<?php

while ($row = mysqli\_fetch\_assoc($consultations\_query)) {

echo "<tr>";

echo "<td>".$row['cons\_date']."</td>";

echo "<td>".$row['cons\_time']."</td>";

echo "<td>".$row['type']."</td>";

echo "<td>".$row['master\_category']."</td>";

echo "<td>".$row['mFullName']."</td>";

echo "</tr>";

}

?>

</table>

</div>

</div>

</body>

</html>

body, html {

height: 100%;

margin: 0;

padding: 0;

background-color: #f5eaea;

overflow: hidden; /\* Prevent scrolling \*/

}

.pcontainer {

max-width: 100%;

margin-left: 100px;

float: left;

overflow: hidden;

height: 100vh; /\* Full viewport height \*/

display: flex;

align-items: center; /\* Center content vertically \*/

justify-content: space-around; /\* Align content to the left \*/

}

.profile {

text-align: center;

max-width: 400px;

display: flex;

flex-direction: column;

align-items: flex-start;

background-color: #b59494; /\* Pastel pink background \*/

border: 2px solid #8b6c7c; /\* Border color \*/

border-radius: 10px; /\* Rounded corners \*/

padding: 20px; /\* Padding inside the profile box \*/

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /\* Optional: Add a subtle shadow for a better look \*/

}

.profile img {

border-radius: 50%;

width: 150px;

height: 150px;

object-fit: cover;

}

.profile h3 {

margin: 10px 0;

color: #000000;

}

.profile p {

margin: 5px 0;

color: #333;

}

.btn, .delete-btn {

display: inline-block;

padding: 10px 20px;

margin: 10px 5px;

color: #ccb5b5;

background: #4e3742;

text-decoration: none;

border-radius: 5px;

}

.btn:hover, .delete-btn:hover {

background: #4e3742;

}

.delete-btn {

margin-top: 20px; /\* Add extra margin to position delete button below others \*/

}

.consultations-container {

max-width: 650px; /\* maximum width \*/

background-color: #b59494; /\* Pastel grey background \*/

padding: 20px; /\* padding inside the container \*/

border-radius: 10px; /\* Rounded corners \*/

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /\* Add a subtle shadow \*/

box-sizing: border-box; /\* include padding in the element's total width and height \*/

}

.consultations-container h2 {

font-size: 18px;

margin-bottom: 10px;

color: #333; /\* Text color \*/

text-align: center; /\* Center the heading \*/

}

.consultations-container table {

width: 100%;

border-collapse: collapse;

}

.consultations-container th,

.consultations-container td {

padding: 8px;

text-align: left;

border: 1px solid #b89797;

border-collapse: collapse;

}

.consultations-container th {

background-color: #d0abab;

}

**//PDF lejupielādes kods**

<?php

require\_once('tcpdf.php');

include 'includes/dbh.inc.php';

session\_start();

$user\_id = $\_SESSION['user\_id'];

if (!isset($user\_id)) {

header('location:login.php');

exit;

}

// Fetch user information

$select = mysqli\_query($conn, "SELECT \* FROM `users` WHERE usersId = '$user\_id'") or die('Query failed');

$fetch = mysqli\_fetch\_assoc($select);

// Fetch user's consultations

$consultations\_query = mysqli\_query($conn, "SELECT consultations.\*, categories.categoryName AS master\_category, masters.mFullName

FROM `consultations`

INNER JOIN `categories\_masters` ON consultations.masterId = categories\_masters.masterId

INNER JOIN `categories` ON categories\_masters.categoryId = categories.categoryId

INNER JOIN `masters` ON consultations.masterId = masters.masterId

WHERE consultations.usersId = '$user\_id'

ORDER BY consultations.cons\_date DESC, consultations.cons\_time DESC") or die('Query failed: ' . mysqli\_error($conn));

// Create new PDF document

$pdf = new TCPDF();

$pdf->AddPage();

// Set document information

$pdf->SetCreator(PDF\_CREATOR);

$pdf->SetAuthor('Jūsu Vietne');

$pdf->SetTitle('Konsultāciju informācija');

$pdf->SetSubject('Konsultāciju detaļas');

$pdf->SetKeywords('TCPDF, PDF, konsultācija, lietotājs');

// Set default header data

$pdf->SetHeaderData(PDF\_HEADER\_LOGO, PDF\_HEADER\_LOGO\_WIDTH, 'Konsultāciju informācija', 'Lietotāja konsultāciju detaļas');

// Set header and footer fonts

$pdf->setHeaderFont(Array(PDF\_FONT\_NAME\_MAIN, '', PDF\_FONT\_SIZE\_MAIN));

$pdf->setFooterFont(Array(PDF\_FONT\_NAME\_DATA, '', PDF\_FONT\_SIZE\_DATA));

// Set default monospaced font

$pdf->SetDefaultMonospacedFont(PDF\_FONT\_MONOSPACED);

// Set margins

$pdf->SetMargins(PDF\_MARGIN\_LEFT, PDF\_MARGIN\_TOP, PDF\_MARGIN\_RIGHT);

$pdf->SetHeaderMargin(PDF\_MARGIN\_HEADER);

$pdf->SetFooterMargin(PDF\_MARGIN\_FOOTER);

// Set auto page breaks

$pdf->SetAutoPageBreak(TRUE, PDF\_MARGIN\_BOTTOM);

// Set image scale factor

$pdf->setImageScale(PDF\_IMAGE\_SCALE\_RATIO);

// Set font

$pdf->SetFont('dejavusans', '', 10);

// Add content

$html = '<h1>Lietotāja informācija</h1>';

$html .= '<p>Vārds: ' . $fetch['name'] . '</p>';

$html .= '<p>E-pasts: ' . $fetch['email'] . '</p>';

$html .= '<p>Telefona numurs: ' . $fetch['number'] . '</p>';

$html .= '<h2>Jūsu konsultācijas</h2>';

$html .= '<table border="1" cellpadding="5">

<thead>

<tr>

<th>Datums</th>

<th>Laiks</th>

<th>Tips</th>

<th>Kategorija</th>

<th>Meistars</th>

</tr>

</thead>

<tbody>';

while ($row = mysqli\_fetch\_assoc($consultations\_query)) {

$html .= '<tr>

<td>' . $row['cons\_date'] . '</td>

<td>' . $row['cons\_time'] . '</td>

<td>' . $row['type'] . '</td>

<td>' . $row['master\_category'] . '</td>

<td>' . $row['mFullName'] . '</td>

</tr>';

}

$html .= '</tbody></table>';

// Output the HTML content

$pdf->writeHTML($html, true, false, true, false, '');

// Close and output PDF document

$pdf->Output('konsultaciju\_informacija.pdf', 'D');

?>

**//Konsultācijas lapa un stils**

<?php

include\_once 'blocks/header.php';

include 'includes/dbh.inc.php';

$error = '';

$successMessage = '';

if (!isset($\_SESSION['user\_id'])) {

echo "Session is not set.<br>";

header("Location: login.php");

exit();

}

// Fetch categories from the database

$categories = [];

$result = $conn->query("SELECT categoryId, categoryName FROM categories");

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

$categories[] = $row;

}

} else {

echo "No categories found.<br>";

}

if (isset($\_SESSION['user\_id'])) {

$userId = $\_SESSION['user\_id'];

$userName = '';

$stmt = $conn->prepare("SELECT name FROM users WHERE usersId = ?");

if ($stmt) {

$stmt->bind\_param("i", $userId);

$stmt->execute();

$stmt->bind\_result($userName);

$stmt->fetch();

$stmt->close();

} else {

echo "Failed to prepare statement.";

}

} else {

header("Location: login.php");

exit();

}

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if (isset($\_POST['cons\_date'], $\_POST['cons\_time'], $\_POST['type'], $\_POST['masterName'])) {

$cons\_date = $\_POST['cons\_date'];

$cons\_time = $\_POST['cons\_time'];

$type = $\_POST['type'];

$masterName = $\_POST['masterName'];

$stmt = $conn->prepare("SELECT masterId FROM masters WHERE mFullName = ?");

if ($stmt) {

$stmt->bind\_param("s", $masterName);

$stmt->execute();

$stmt->bind\_result($masterId);

$stmt->fetch();

$stmt->close();

} else {

$error = "Failed to prepare statement.";

}

if ($masterId) {

$stmt = $conn->prepare("SELECT COUNT(\*) FROM consultations WHERE cons\_date = ? AND cons\_time = ? AND masterId = ?");

if ($stmt) {

$stmt->bind\_param("ssi", $cons\_date, $cons\_time, $masterId);

$stmt->execute();

$stmt->bind\_result($count);

$stmt->fetch();

$stmt->close();

} else {

$error = "Failed to prepare statement.";

}

if ($count == 0) {

$stmt = $conn->prepare("INSERT INTO consultations (cons\_date, cons\_time, type, masterId, userName, usersId) VALUES (?, ?, ?, ?, ?, ?)");

if ($stmt) {

$stmt->bind\_param("sssisi", $cons\_date, $cons\_time, $type, $masterId, $userName, $userId);

if ($stmt->execute()) {

$successMessage = "Consultation added successfully";

} else {

$error = "Error: " . $stmt->error;

}

$stmt->close();

} else {

$error = "Failed to prepare statement.";

}

} else {

$error = "The selected date and time is already booked.";

}

} else {

$error = "Invalid master selected.";

}

} else {

$error = "All fields are required.";

}

}

$conn->close();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/consultations.css">

<title>Pievienot konsultāciju</title>

<script>

function disablePastDatesAndTimes() {

var today = new Date().toISOString().split('T')[0];

document.getElementById("cons\_date").setAttribute("min", today);

}

function generateTimeOptions() {

var timeSelect = document.getElementById("cons\_time");

var startTime = 9; // 9:00

var endTime = 19; // 19:00

for (var hour = startTime; hour <= endTime; hour++) {

["00", "30"].forEach(function (minutes) {

var timeValue = (hour < 10 ? "0" : "") + hour + ":" + minutes;

var option = document.createElement("option");

option.value = timeValue;

option.text = timeValue;

timeSelect.appendChild(option);

});

}

}

function fetchMasters(categoryId) {

var xhr = new XMLHttpRequest();

xhr.open("POST", "fetch\_masters.php", true);

xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

xhr.onreadystatechange = function () {

if (xhr.readyState === 4 && xhr.status === 200) {

document.getElementById("masterName").innerHTML = xhr.responseText;

fetchBookedTimes();

}

};

xhr.send("categoryId=" + categoryId);

}

function fetchBookedTimes() {

var date = document.getElementById("cons\_date").value;

var masterId = document.getElementById("masterName").value;

if (!date || !masterId) return;

var xhr = new XMLHttpRequest();

xhr.open("POST", "fetch\_booked\_times.php", true);

xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

xhr.onreadystatechange = function () {

if (xhr.readyState === 4 && xhr.status === 200) {

var bookedTimes = JSON.parse(xhr.responseText);

var timeSelect = document.getElementById("cons\_time");

var options = timeSelect.getElementsByTagName("option");

for (var i = 0; i < options.length; i++) {

options[i].disabled = bookedTimes.includes(options[i].value);

}

}

};

xhr.send("date=" + date + "&masterId=" + masterId);

}

window.onload = function() {

disablePastDatesAndTimes();

generateTimeOptions();

setInterval(fetchBookedTimes, 5400000);

}

</script>

</head>

<body>

<div class="intro-text">

<h2>Laipni lūdzam mūsu konsultāciju sadaļā!</h2>

<p>

Mēs piedāvājam plašu pakalpojumu klāstu, kas vērsti uz jūsu personīgo aprūpi un labklājību.

Mūsu meistari ir ļoti atbildīgi un profesionāli, nodrošinot visaugstāko apkalpošanas kvalitāti.

Jūs varat pierakstīties uz konsultāciju, izvēloties sev ērtāko datumu un laiku, kā arī piemērotāko meistaru.

Neatkarīgi no tā, vai vēlaties klātienes tikšanos vai attālinātu konsultāciju, mēs esam šeit, lai jums palīdzētu!

</p>

</div>

<form action="consultation.php" method="post">

<h2>Pieraksts uz konsultāciju</h2>

<label for="cons\_date">Konsultācijas datums</label>

<input type="date" id="cons\_date" name="cons\_date" onchange="fetchBookedTimes()" required><br><br>

<label for="cons\_time">Konsultācijas laiks</label>

<select id="cons\_time" name="cons\_time" required>

<option value="">Laiks</option>

</select><br><br>

<label for="type">Konsultācijas veids</label>

<select id="type" name="type" required>

<option value="">Veids</option>

<option value="Attalināti">Attalināti</option>

<option value="Klātienē">Klātienē</option>

</select><br><br>

<label for="category">Kategorija:</label>

<select id="category" name="category" onchange="fetchMasters(this.value)" required>

<option value="">Kategorija</option>

<?php foreach ($categories as $category): ?>

<option value="<?php echo htmlspecialchars($category['categoryId']); ?>">

<?php echo htmlspecialchars($category['categoryName']); ?>

</option>

<?php endforeach; ?>

</select><br><br>

<label for="masterName">Meistars:</label>

<select id="masterName" name="masterName" onchange="fetchBookedTimes()" required>

<option value="">Meistars</option>

</select><br><br>

<?php if ($error): ?>

<p style="color: grey;"><?php echo $error; ?></p>

<?php endif; ?>

<?php if ($successMessage): ?>

<p style="color: green;"><?php echo $successMessage; ?></p>

<?php endif; ?>

<input type="submit" value="Akceptēt pierakstu">

</form>

</body>

</html>

body {

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.intro-text {

background: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

max-width: 600px;

margin: 20px auto;

text-align: center;

}

.intro-text h2 {

color: #333;

}

.intro-text p {

color: #555;

line-height: 1.6;

}

form {

background: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

max-width: 330px;

margin: 20px auto;

}

h2 {

text-align: center;

color: #333;

}

label {

display: block;

margin: 10px 0 5px;

color: #555;

}

input[type="date"],

input[type="time"],

input[type="number"],

input[type="text"],

input[type="submit"],

select {

width: 100%;

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

input[type="submit"] {

background: #b85c7e;

color: #fff;

border: none;

cursor: pointer;

font-size: 16px;

}

input[type="submit"]:hover {

background: #7c3d6d;

}

**//Administratora lapa un stils priekš meistaru pievienošanas**

<?php

include "adminheader.php";

include '../includes/dbh.inc.php';

$successMessage = $errorMessage = "";

// Iegūstam kategoriju sarakstu no datu bāzes

$categoryListQuery = "SELECT \* FROM categories";

$categoryListResult = $conn->query($categoryListQuery);

// Iegūstam meistaru sarakstu no datu bāzes

$masterListQuery = "SELECT masters.masterId, masters.mFullName, categories.categoryName, masters.workTimeOpen, masters.workTimeClose

FROM masters

INNER JOIN categories\_masters ON masters.masterId = categories\_masters.masterId

INNER JOIN categories ON categories\_masters.categoryId = categories.categoryId";

$masterListResult = $conn->query($masterListQuery);

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['confirm\_delete'])) {

$masterId = $\_POST['master\_id'];

$confirmDelete = $\_POST['confirm\_delete'];

if ($confirmDelete == 'yes') {

// Dzēšanas vaicājums kategorijām un meistariem

$deleteCategoriesMastersQuery = "DELETE FROM categories\_masters WHERE masterId = ?";

$stmt = $conn->prepare($deleteCategoriesMastersQuery);

$stmt->bind\_param("i", $masterId);

$stmt->execute();

// Dzēšanas vaicājums no tabulas "masters"

$deleteMasterQuery = "DELETE FROM masters WHERE masterId = ?";

$stmt = $conn->prepare($deleteMasterQuery);

$stmt->bind\_param("i", $masterId);

if ($stmt->execute()) {

$successMessage = "Meistars veiksmīgi dzēsts.";

} else {

$errorMessage = "Dzēšanas kļūda: " . $conn->error;

}

}

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && !isset($\_POST['confirm\_delete'])) {

// Jaunās meistara pievienošanas apstrāde

$mFullName = $\_POST['mFullName'];

$categoryName = $\_POST['category']; // Pārņemam izvēlēto kategoriju

$workTimeOpen = $\_POST['workTimeOpen'];

$workTimeClose = $\_POST['workTimeClose'];

$mEmail = $\_POST['mEmail'];

$mPassword = trim($\_POST['mPassword']);

// Paroles hash iegūšana

$hashed\_password = password\_hash($mPassword, PASSWORD\_DEFAULT);

// Attēla augšupielādes apstrāde

$targetDir = dirname(\_\_FILE\_\_) . "/masteruploads/"; // Pielāgots mērķa katalogs

$targetFile = $targetDir . basename($\_FILES["photo"]["name"]);

$uploadOk = 1;

$imageFileType = strtolower(pathinfo($targetFile, PATHINFO\_EXTENSION));

// Pārbauda, vai augšupielādētais fails ir attēls

$check = getimagesize($\_FILES["photo"]["tmp\_name"]);

if($check !== false) {

$uploadOk = 1;

} else {

$errorMessage = "Fails nav attēls.";

$uploadOk = 0;

}

// Pārbauda, vai fails jau eksistē

if (file\_exists($targetFile)) {

$errorMessage = "Atvainojiet, fails jau eksistē.";

$uploadOk = 0;

}

// Pārbauda attēla izmēru

if ($\_FILES["photo"]["size"] > 500000) {

$errorMessage = "Atvainojiet, jūsu fails ir pārāk liels.";

$uploadOk = 0;

}

// Atļautie attēla formāti

if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"

&& $imageFileType != "gif" && $imageFileType != "jfif") {

$errorMessage = "Atvainojiet, atļauti tikai JPG, JPEG, PNG, GIF un JFIF faili.";

$uploadOk = 0;

}

// Pārbauda, vai $uploadOk nav iestatīts uz 0 kā kļūda

if ($uploadOk == 0) {

$errorMessage = "Atvainojiet, jūsu fails netika augšupielādēts.";

} else {

// ja viss kārtībā, mēģiniet augšupielādēt failu

if (move\_uploaded\_file($\_FILES["photo"]["tmp\_name"], $targetFile)) {

// Fails veiksmīgi augšupielādēts, tagad iegūstam kategorijas ID no tabulas 'categories'

// Pārbauda, vai kategorija pastāv

$catQuery = "SELECT categoryId FROM categories WHERE categoryName = ?";

$stmt = $conn->prepare($catQuery);

$stmt->bind\_param("s", $categoryName);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

$row = $result->fetch\_assoc();

$categoryId = $row["categoryId"];

// Kategorijas ID ir iegūts, tagad ievietojam datus tabulā 'masters'

$query = "INSERT INTO masters (mFullName, mPhoto, workTimeOpen, workTimeClose, mEmail, mPassword, type) VALUES (?, ?, ?, ?, ?, ?, ?)";

$stmt = $conn->prepare($query);

$type = 2; // Здесь вносим значение 2 для поля type

$stmt->bind\_param("ssssssi", $mFullName, $targetFile, $workTimeOpen, $workTimeClose, $mEmail, $hashed\_password, $type);

if ($stmt->execute()) {

$masterId = $conn->insert\_id;

// Tagad pievienojam meistara ID tabulā 'categories\_masters'

$insertQuery = "INSERT INTO categories\_masters (categoryId, masterId) VALUES (?, ?)";

$stmt = $conn->prepare($insertQuery);

$stmt->bind\_param("ii", $categoryId, $masterId);

if ($stmt->execute()) {

$successMessage = "Jaunais meistars veiksmīgi pievienots";

} else {

$errorMessage = "Kļūda: " . $insertQuery . "<br>" . $conn->error;

}

} else {

$errorMessage = "Kļūda: " . $query . "<br>" . $conn->error;

}

} else {

$errorMessage = "Izvēlētā kategorija nav atrasta datu bāzē.";

}

$stmt->close();

} else {

$errorMessage = "Atvainojiet, radās kļūda, augšupielādējot jūsu failu.";

}

}

}

?>

<!DOCTYPE html>

<html lang="lv">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Pievienot meistaru</title>

<link rel="stylesheet" href="css/adminmasters.css">

<script>

function confirmDelete() {

return confirm("Vai tiešām vēlaties dzēst meistaru?");

}

</script>

</head>

<body>

<div class="container">

<!-- Forma, lai pievienotu jaunu meistaru -->

<div class="mmcontainer">

<h2>Pievienot jaunu meistaru</h2>

<?php if (!empty($errorMessage)) : ?>

<div class="error"><?php echo $errorMessage; ?></div>

<?php endif; ?>

<?php if (!empty($successMessage)) : ?>

<div class="success"><?php echo $successMessage; ?></div>

<?php endif; ?>

<form action="adminmasters.php" method="post" enctype="multipart/form-data">

<label for="mFullName">Pilnais vārds:</label>

<input type="text" id="mFullName" name="mFullName" required><br><br>

<label for="photo">Bilde:</label>

<input type="file" id="photo" name="photo" accept="image/\*" required><br><br>

<label for="category">Kategorija:</label>

<select id="category" name="category" required>

<option value="" selected disabled>Izvēlieties kategoriju</option>

<?php

if ($categoryListResult->num\_rows > 0) {

while($row = $categoryListResult->fetch\_assoc()) {

echo "<option value='".$row["categoryName"]."'>".$row["categoryName"]."</option>";

}

}

?>

</select><br><br>

<label for="workTimeOpen">Darba sākuma laiks:</label>

<input type="time" id="workTimeOpen" name="workTimeOpen" required><br><br>

<label for="workTimeClose">Darba beigu laiks:</label>

<input type="time" id="workTimeClose" name="workTimeClose" required><br><br>

<label for="mEmail">E-pasta adrese:</label>

<input type="email" id="mEmail" name="mEmail" required><br><br>

<label for="mPassword">Parole:</label>

<input type="password" id="mPassword" name="mPassword" required><br><br>

<button type="submit">Pievienot meistaru</button>

</form>

</div>

<!-- Tabula ar esošajiem meistariem -->

<div class="mmcontainer">

<h2>Esošie meistari</h2>

<table>

<tr>

<th>ID</th>

<th>Pilnais vārds</th>

<th>Kategorija</th>

<th>Darba laiks</th>

<th>Darbība</th>

</tr>

<?php

if ($masterListResult->num\_rows > 0) {

while($row = $masterListResult->fetch\_assoc()) {

echo "<tr>";

echo "<td>".$row["masterId"]."</td>";

echo "<td>".$row["mFullName"]."</td>";

echo "<td>".$row["categoryName"]."</td>";

echo "<td>".$row["workTimeOpen"]. " - " . $row["workTimeClose"]."</td>";

echo "<td>

<form action='adminmasters.php' method='post' onsubmit='return confirmDelete();'>

<input type='hidden' name='master\_id' value='".$row["masterId"]."'>

<input type='hidden' name='confirm\_delete' value='yes'>

<button type='submit'>Dzēst</button>

</form>

</td>";

echo "</tr>";

}

}

?>

</table>

</div>

</div>

</body>

</html>

body {

padding-top: 50px; /\* Augšpusē izmanto fiksēto augšējo joslu \*/

}

header {

position: fixed;

top: 0;

left: 0;

width: 100%;

background-color: #333;

color: #fff;

padding: 10px 0;

text-align: center;

}

.container {

margin-top: 50px; /\* Atstāj vietu fiksētajam header un formas virsū \*/

display: flex;

justify-content: space-around;

}

.container > div {

flex: 1;

margin: 0 20px;

}

.mmcontainer, .mccontainer {

max-width: 330px;

margin: 50px;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

background-color: #f9f9f9;

}

.mccontainer {

max-height: fit-content;

}

h2 {

margin-top: 0;

text-align: center;

}

form {

margin-top: 20px;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"], input[type="file"] {

width: 100%;

padding: 8px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 3px;

box-sizing: border-box;

}

button[type="submit"] {

background-color: #af4c81;

color: white;

padding: 10px 20px;

border: none;

border-radius: 3px;

cursor: pointer;

}

button[type="submit"]:hover {

background-color: #894861;

}

.error {

color: rgb(35, 24, 24);

margin-top: 5px;

}

**//Lietotāja profila iestatījumu rediģēšanas lapa un stils**

<?php

include 'includes/dbh.inc.php';

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header('Location: login.php');

exit;

}

$user\_id = $\_SESSION['user\_id'];

$select = mysqli\_query($conn, "SELECT \* FROM `users` WHERE usersId = '$user\_id'") or die('Query failed: ' . mysqli\_error($conn));

$fetch = mysqli\_fetch\_assoc($select);

if (isset($\_POST['update\_profile'])) {

$update\_name = mysqli\_real\_escape\_string($conn, $\_POST['update\_name']);

$update\_email = mysqli\_real\_escape\_string($conn, $\_POST['update\_email']);

$update\_number = mysqli\_real\_escape\_string($conn, $\_POST['update\_number']); // Add this line for updating the phone number

// Update name, email, and number

mysqli\_query($conn, "UPDATE `users` SET name = '$update\_name', email = '$update\_email', number = '$update\_number' WHERE usersId = '$user\_id'") or die('Query failed: ' . mysqli\_error($conn));

$old\_pass\_input = mysqli\_real\_escape\_string($conn, $\_POST['update\_pass']);

$new\_pass = mysqli\_real\_escape\_string($conn, $\_POST['new\_pass']);

$confirm\_pass = mysqli\_real\_escape\_string($conn, $\_POST['confirm\_pass']);

// Fetch the current password from the database

$result = mysqli\_query($conn, "SELECT password FROM `users` WHERE usersId = '$user\_id'") or die('Query failed: ' . mysqli\_error($conn));

$row = mysqli\_fetch\_assoc($result);

$current\_password\_hash = $row['password'];

// Verify old password

if (!empty($old\_pass\_input) && password\_verify($old\_pass\_input, $current\_password\_hash)) {

if (!empty($new\_pass) && !empty($confirm\_pass)) {

if ($new\_pass != $confirm\_pass) {

$message[] = 'Confirm password not matched!';

} else {

// Validate new password strength

if (strlen($new\_pass) < 8 || !preg\_match("#[0-9]+#", $new\_pass) || !preg\_match("#[A-Z]+#", $new\_pass) || !preg\_match("#[a-z]+#", $new\_pass) || !preg\_match("#[\W]+#", $new\_pass)) {

$message[] = 'New password does not meet the criteria.';

} else {

$hashed\_new\_pass = password\_hash($new\_pass, PASSWORD\_DEFAULT);

mysqli\_query($conn, "UPDATE `users` SET password = '$hashed\_new\_pass' WHERE usersId = '$user\_id'") or die('Query failed: ' . mysqli\_error($conn));

$message[] = 'Password updated successfully!';

}

}

} elseif (!empty($new\_pass) || !empty($confirm\_pass)) {

$message[] = 'Both new password and confirm password must be filled!';

}

} else {

if (!empty($old\_pass\_input)) {

$message[] = 'Old password not matched!';

}

}

// Image update logic

if (isset($\_FILES['update\_image']) && $\_FILES['update\_image']['error'] === UPLOAD\_ERR\_OK) {

$update\_image\_name = $\_FILES['update\_image']['name'];

$update\_image\_size = $\_FILES['update\_image']['size'];

$update\_image\_tmp\_name = $\_FILES['update\_image']['tmp\_name'];

$update\_image\_folder = 'uploaded\_img/' . $update\_image\_name;

if ($update\_image\_size > 2000000) {

$message[] = 'Image is too large';

} else {

$image\_update\_query = mysqli\_query($conn, "UPDATE `users` SET image = '$update\_image\_name' WHERE usersId = '$user\_id'") or die('Query failed: ' . mysqli\_error($conn));

if ($image\_update\_query) {

move\_uploaded\_file($update\_image\_tmp\_name, $update\_image\_folder);

$message[] = 'Image updated successfully!';

}

}

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Update Profile</title>

<link rel="stylesheet" href="css/update\_profile.css">

</head>

<body>

<div class="update-profile">

<form action="" method="post" enctype="multipart/form-data">

<?php

if (isset($fetch) && $fetch['image'] == '') {

echo '<img src="images/default-avatar.png">';

} elseif (isset($fetch)) {

echo '<img src="uploaded\_img/' . $fetch['image'] . '">';

}

if (isset($message)) {

foreach ($message as $msg) {

echo '<div class="message">' . $msg . '</div>';

}

}

?>

<div class="flex">

<div class="inputBox">

<span>Username :</span>

<input type="text" name="update\_name" value="<?php echo htmlspecialchars($fetch['name']); ?>" class="box">

<span>Email :</span>

<input type="email" name="update\_email" value="<?php echo htmlspecialchars($fetch['email']); ?>" class="box">

<span>Phone Number :</span> <!-- Add this line -->

<input type="text" name="update\_number" value="<?php echo htmlspecialchars($fetch['number']); ?>" class="box"> <!-- Add this line -->

<span>Update your pic :</span>

<input type="file" name="update\_image" accept="image/jpg, image/jpeg, image/png" class="box">

</div>

<div class="inputBox">

<span>Old password :</span>

<input type="password" name="update\_pass" placeholder="Previous password" class="box">

<span>New password :</span>

<input type="password" name="new\_pass" placeholder="New password" class="box">

<span>Confirm password :</span>

<input type="password" name="confirm\_pass" placeholder="Confirm new password" class="box">

</div>

</div>

<div class="btn-container">

<input type="submit" value="Update profile" name="update\_profile" class="btn">

<a href="profile.php" class="delete-btn">Go back</a>

</div>

</form>

</div>

</body>

</html>

body, html {

height: 100%;

margin: 0;

padding: 0;

background-color: #f0f0f0;

display: flex;

align-items: center;

justify-content: center;

}

.update-profile {

width: 500px;

background: #fff;

padding: 30px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

border-radius: 10px;

text-align: center;

}

.update-profile img {

border-radius: 50%;

width: 150px;

height: 150px;

object-fit: cover;

margin-bottom: 20px;

}

.update-profile .flex {

display: flex;

flex-direction: column;

gap: 20px;

margin-bottom: 20px;

}

.update-profile .inputBox {

display: flex;

flex-direction: column;

gap: 10px;

}

.update-profile .inputBox span {

text-align: left;

color: #333;

font-weight: bold;

}

.update-profile .box {

width: 100%;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

box-sizing: border-box;

}

.update-profile .btn-container {

display: flex;

justify-content: space-between;

gap: 10px;

}

.update-profile .btn, .delete-btn {

width: 48%; /\* Ensure both buttons take up the same width within the container \*/

padding: 10px;

color: #fff;

background: #b85272;

text-decoration: none;

border-radius: 5px;

border: none;

outline: none;

transition: background-color 0.3s ease;

text-align: center;

}

.update-profile .btn:hover, .delete-btn:hover {

background: #dc658b;

}

.update-profile .delete-btn {

background: #b85272;

}

.update-profile .message {

margin: 10px 0;

padding: 10px;

background: #f8d7da;

color: #721c24;

border: 1px solid #f5c6cb;

border-radius: 5px;

}

**//Administratora lapa lai apskatītu produktus tabulas veidā**

<?php

include "adminheader.php";

?>

<?php

require\_once "../includes/dbh.inc.php";

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

if (!$conn) {

die("Connect error: " . mysqli\_connect\_error());

}

$uploadDirectory = "./uploads";

chmod($uploadDirectory, 0755);

// Fetching product information from database

$sql = "SELECT productId, name, size, description, price, image FROM products";

$result = mysqli\_query($conn, $sql);

// Checking for results

if (mysqli\_num\_rows($result) > 0) {

// Data output from table HTML

echo "<table>

<tr>

<th>ID</th>

<th>Image</th>

<th>Name</th>

<th>Size</th>

<th>Description</th>

<th>Price</th>

<th>Update</th>

<th>Delete</th>

</tr>";

while($row = mysqli\_fetch\_assoc($result)) {

echo "<tr>

<td>".$row["productId"]."</td>

<td><img src='".$row["image"]."' alt='Product Image' width='100'></td>

<td>".$row["name"]."</td>

<td>".$row["size"]."</td>

<td>".$row["description"]."</td>

<td>$".$row["price"]."</td>

<td><button class='tbb' onclick=\"window.location.href='update.php?productId=".$row["productId"]."'\">Update</button></td>

<td><button class='tbb' onclick=\"deleteProduct(".$row["productId"].")\">Delete</button></td>

</tr>";

}

echo "</table>";

} else {

echo "No result";

}

// Closing the database connection

mysqli\_close($conn);

?>

<!-- Add a "Create Product" button -->

<div class="create-product-button">

<button onclick="window.location.href='create.php'">Create Product</button>

</div>

<link href="css/storeupdate.css" rel="stylesheet">

<script>

// JavaScript function to handle the product deletion

function deleteProduct(productId) {

if (confirm("Are you sure you want to delete this product?")) {

// Redirect to the delete page or perform an AJAX request to delete the product

window.location.href = "delete.php?productId=" + productId;

}

}

</script>

**//Administratora lapa priekš produkta pievienošanas un stils (apvienots fails ar administratora produktu lapu un produktu rediģēšanas lapu)**

<?php

require\_once "../includes/dbh.inc.php"; // Include the database connection file

include\_once "adminheader.php";

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Retrieve the product details from the form

$name = $\_POST["name"];

$size = $\_POST["size"];

$description = $\_POST["description"];

$price = $\_POST["price"];

// Handle file upload

$imagePath = uploadImage(); // Function to upload the image and return its path

if ($imagePath !== false) {

// Convert the price to a valid format (float)

$price = convertPriceToFloat($price);

// Insert the product into the database with the image path

$sql = "INSERT INTO products (name, size, description, price, image) VALUES (?, ?, ?, ?, ?, ?)";

if ($stmt = mysqli\_prepare($conn, $sql)) {

// Bind the parameters

mysqli\_stmt\_bind\_param($stmt, "ssssds", $name, $size, $description, $price, $imagePath);

// Execute the statement

if (mysqli\_stmt\_execute($stmt)) {

// Get the ID of the newly inserted product

$product\_id = mysqli\_insert\_id($conn);

// Close the statement

mysqli\_stmt\_close($stmt);

} else {

echo "Error: " . mysqli\_error($conn);

}

} else {

echo "Error: " . mysqli\_error($conn);

}

// Close the database connection

mysqli\_close($conn);

} else {

echo "Error uploading the image.";

}

}

// Function to convert the input price to a float

function convertPriceToFloat($price) {

// Replace any commas with dots

$cleaned\_price = str\_replace(',', '.', $price);

// Convert to a float

return floatval($cleaned\_price);

}

// Function to handle file upload and return the image path

function uploadImage() {

// Check if a file was uploaded

if (isset($\_FILES["image"]) && $\_FILES["image"]["error"] == UPLOAD\_ERR\_OK) {

$targetDirectory = "uploads/"; // Specify the directory where you want to store uploaded images

// Generate a unique file name to avoid overwriting

$imagePath = $targetDirectory . uniqid() . "\_" . $\_FILES["image"]["name"];

// Move the uploaded file to the target directory

if (move\_uploaded\_file($\_FILES["image"]["tmp\_name"], $imagePath)) {

return $imagePath;

}

}

return false; // Return false if there was an error or no file was uploaded

}

?>

<link href="css/update.css" rel="stylesheet">

<div class="blnk"></div>

<!-- HTML form to enter the product details -->

<form method="POST" action="" enctype="multipart/form-data">

<label for="image">Image:</label>

<input type="file" name="image" accept="image/\*">

<label for="name">Name:</label>

<input type="text" name="name" required><br>

<label for="size">Size:</label>

<input type="text" name="size" required><br>

<label for="description">Description:</label>

<textarea name="description" required></textarea><br>

<label for="price">Price:</label>

<input type="text" name="price" required><br>

<input type="submit" value="Create Product">

</form>

<div class="blnk"></div>

table {

width: 100%;

border-collapse: collapse;

border: 5px solid rgb(78, 57, 57);

}

th {

letter-spacing: 2px;

border-collapse: collapse;

border: 2px solid rgb(78, 57, 57);

}

td {

letter-spacing: 1px;

border-collapse: collapse;

border: 2px solid rgb(78, 57, 57);

}

.create-product-button {

text-align: center;

}

.create-product-button button {

color: #531024 !important;

border: 2px solid #000000;

letter-spacing: 0px;

font-weight: 900;

text-transform: uppercase;

margin-top: 30px;

background-color: transparent !important;

stroke: none;

position: relative;

padding: 9px;

}

.tbb {

color: #a45f74 !important;

border: 2px solid #a45f74;

border-radius: 15px;

letter-spacing: 0px;

font-weight: 700;

text-transform: uppercase;

background-color: transparent !important;

stroke: none;

position: relative;

padding: 7px;

margin: 5px;

}

**//Produkta rediģēšanas lapa**

<?php

require\_once "../includes/dbh.inc.php";

include\_once "adminheader.php";

// Check if product ID is provided in URL

$product\_id = isset($\_GET['productId']) ? $\_GET['productId'] : '';

// Prepare the SQL statement with a placeholder for the product ID

$sql = "SELECT \* FROM `products` WHERE `productId` = ?";

$stmt = mysqli\_prepare($conn, $sql);

mysqli\_stmt\_bind\_param($stmt, "i", $product\_id);

mysqli\_stmt\_execute($stmt);

// Get the result set

$result = mysqli\_stmt\_get\_result($stmt);

// If the product exists, display the product information in a form

if (mysqli\_num\_rows($result) > 0) {

$row = mysqli\_fetch\_assoc($result);

?>

<link href="css/update.css" rel="stylesheet">

<div class="blnk"></div>

<form action="<?php echo htmlspecialchars($\_SERVER['PHP\_SELF']); ?>" method="post" enctype="multipart/form-data">

<input type="hidden" name="product\_id" value="<?php echo htmlspecialchars($row['productId']); ?>">

<label for="product\_name">Product Name:</label>

<input type="text" name="product\_name" value="<?php echo htmlspecialchars($row['name']); ?>">

<label for="product\_size">Product Size:</label>

<input type="text" name="product\_size" value="<?php echo htmlspecialchars($row['size']); ?>">

<label for="product\_description">Product Description:</label>

<textarea name="product\_description"><?php echo htmlspecialchars($row['description']); ?></textarea>

<label for="product\_price">Product Price:</label>

<input type="text" name="product\_price" value="<?php echo htmlspecialchars($row['price']); ?>">

<label for="product\_image">Product Image:</label>

<input type="file" name="product\_image">

<input type="submit" value="Update" name="update\_product">

</form>

<div class="blnk"></div>

<?php

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["update\_product"])) {

if (isset($\_POST["product\_id"])) {

$product\_id = $\_POST["product\_id"];

$product\_name = $\_POST["product\_name"];

$product\_size = $\_POST["product\_size"];

$product\_description = $\_POST["product\_description"];

$product\_price = validateAndConvertPrice($\_POST["product\_price"]);

// Handle image upload only if a file was selected

if (!empty($\_FILES["product\_image"]["name"])) {

$uploadDirectory = "uploads/";

$uploadedFile = $\_FILES["product\_image"]["tmp\_name"];

$product\_image = $uploadDirectory . $\_FILES["product\_image"]["name"];

if ($\_FILES["product\_image"]["error"] !== UPLOAD\_ERR\_OK) {

echo "Error uploading the image. Error code: " . $\_FILES["product\_image"]["error"];

exit; // Exit the script to prevent further execution

}

if (!move\_uploaded\_file($uploadedFile, $product\_image)) {

echo "Error moving the uploaded file.";

exit;

}

}

// Update the product information in the database, including the image file path

$sql = "UPDATE `products` SET `name` = ?, `size` = ?, `description` = ?, `price` = ?, `image` = ? WHERE `productId` = ?";

$stmt = mysqli\_prepare($conn, $sql);

mysqli\_stmt\_bind\_param($stmt, "sssdisi", $product\_name, $product\_size, $product\_description, $product\_price, $product\_image, $product\_id);

if (mysqli\_stmt\_execute($stmt)) {

echo "Product updated successfully.";

} else {

echo "Error updating product: " . mysqli\_error($conn);

}

mysqli\_stmt\_close($stmt);

}

}

// Function to validate and convert the product price

function validateAndConvertPrice($price) {

// Remove any non-numeric characters and then convert to a float

$cleaned\_price = filter\_var($price, FILTER\_SANITIZE\_NUMBER\_FLOAT, FILTER\_FLAG\_ALLOW\_FRACTION);

return floatval($cleaned\_price);

}

// Close the prepared statement and the database connection

mysqli\_close($conn);

?>

**//Lapa ar informāciju par meistariem**

<!DOCTYPE html>

<html lang="ru">

<?php include\_once 'blocks/header.php'; ?>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" type="text/css" href="/css/masters.css">

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

</head>

<body>

<div>

<section class="aboutmasters">

<a></a>

</section>

<section class="m1">

<?php

// Вывод информации о мастерах

include 'includes/dbh.inc.php';

$sql = "SELECT m.masterId, m.mFullName, m.mPhoto, c.categoryName, m.workTimeOpen, m.workTimeClose

FROM masters AS m

INNER JOIN categories\_masters AS cm ON m.masterId = cm.masterId

INNER JOIN categories AS c ON cm.categoryId = c.categoryId

GROUP BY m.masterId

ORDER BY m.mFullName";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

$mFullName = $row['mFullName'];

$mPhoto = $row['mPhoto'];

$categoryName = $row['categoryName'];

$workTimeOpen = $row['workTimeOpen'];

$workTimeClose = $row['workTimeClose'];

$imagePath = "../admin/masteruploads/" . basename($mPhoto);

echo "<div class='master'>

<img src='$imagePath' alt='$mFullName'>

<h2>$mFullName</h2>

<p>$categoryName</p>

<p>Work Time: $workTimeOpen-$workTimeClose</p>

</div>";

}

} else {

echo "No masters found.";

}

$conn->close();

?>

</section>

</div>

</body>

</html>