

RĪGAS VALSTS TEHNIKUMS

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

KVALIFIKĀCIJAS DARBS

“Baletskolas kostīmu datu uzskaites sistēma”

Paskaidrojošais raksts 120 lpp.

Audzēknis:

Katrīna Gandzjuka

Prakses vadītājs:

Ilona Demčenko

Nodaļas vadītājs:

Normunds Barbāns

Rīga 2024

ANOTĀCIJA

Baletskolas kostīmu uzskaites sistēma ir izstrādāta, lai efektīvi pārvaldītu skolēnu, vecāku un pedagogu datus, kā arī organizētu un vadītu koncertus un to numurus. Sistēmas izstrādes laikā tika izmantotas tādas tehnoloģijas kā HTML5, CSS Bootstrap 3.4, JavaScript, PHP 8.1, Laravel 10 un MySQL. Papildus tika izmantota DomPDF bibliotēka, lai ģenerētu PDF dokumentus no HTML un CSS koda.

Kvalifikācijas darbs ietver ievadu, uzdevumu nostādni, prasību specifikāciju, uzdevuma risināšanas līdzekļu izvēles pamatojumu, programmatūras produkta modelēšanas un projektēšanas aprakstu, datu struktūru aprakstu un secinājumus. Ievadā ir aprakstīta baletskolas kostīmu uzskaites sistēmas aktualitāte. Uzdevumu nostādnē ir norādīti uzdevumi, kurus sistēmai būs nepieciešams veikt. Prasību specifikācija sastāv no ieejas un izejas informācijas, kā arī no sistēmas funkcionālajām un nefunkcionālajām prasībām. Uzdevuma risināšanas līdzekļu izvēles pamatojumā ir norādīti izstrādāšanai izmantotie līdzekļi un to pielietojums. Programmatūras produkta modelēšanas un projektēšanas apraksts ietver sistēmas struktūras modeli un funkcionālo sistēmas modeli. Datu struktūru aprakstā tiek parādīta datu bāzes relāciju shēma, kā arī tabulu struktūra ar datu tipu un datu garuma norādīšanu. Lietotāja ceļvedī ir norādītas sistēmas prasības aparatūrai un programmatūrai, instalācija un palaišana, kā arī programmas lietošanas pamācība. Testa piemērā ir dots detalizēts autentificēšanās un jauna konta izveides apraksts ar vizuāliem attēliem.

Kvalifikācijas darbs sastāv no 120 lappusēm, kurās ietilpst 81 attēli, 15 tabulas un 1 pielikuma. Pielikums satur programmas pirmkodu fragmentus.

ANNOTATION

The ballet school costume management system is designed to efficiently manage data of students, parents, and teachers, as well as to organize and manage concerts and their performances. Technologies such as HTML5, CSS Bootstrap 3.4, JavaScript, PHP 8.1, Laravel 10, and MySQL were used during the system's development. Additionally, the DomPDF library was used to generate PDF documents from HTML and CSS code.

The qualification thesis includes an introduction, task statement, requirements specification, justification for the choice of tools, description of software product modeling and design, description of data structures, and conclusions. The introduction describes the relevance of the ballet school costume management system. The task statement outlines the tasks that the system needs to perform. The requirements specification consists of input and output information, as well as the system's functional and non-functional requirements. The justification for the choice of tools details the tools used for development and their purposes. The software product modeling and design description includes the system's structural model and functional system model. The description of data structures shows the database relational schema and table structures, including data types and lengths. The user guide specifies the system's hardware and software requirements, installation and launch instructions, and usage instructions. The test example provides a detailed description of the authentication and account creation processes with visual illustrations.

The qualification thesis consists of 120 pages, including 81 images, 15 tables, and 1 appendice. The appendice contains code fragments.

SATURS

IEVADS	5
1. UZDEVUMA NOSTĀDNE	7
2. PRASĪBU SPECIFIKĀCIJA.....	9
2.1. Ieejas un izejas informācijas apraksts.....	9
2.1.1. Ieejas informācijas apraksts	9
2.1.2. Izejas informācijas apraksts	11
2.2. Funkcionālās prasības.....	12
2.3. Nefunkcionālās prasības	17
3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS	21
4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA	22
4.1. Sistēmas struktūras modelis.....	22
4.1.1. Sistēmas arhitektūra	22
4.1.2. Sistēmas ER modelis.....	23
4.2. Funkcionālais sistēmas modelis.....	24
4.2.1. Datu plūsmu modelis	24
5. DATU STRUKTŪRU APRAKSTS	27
6. LIETOTĀJA CEĻVEDIS.....	34
6.1. Sistēmas prasības aparatūrai un programmatūrai	34
6.2. Sistēmas instalācija un palaišana	34
6.3. Programmas apraksts	36
6.4. Testa piemērs.....	66
NOBEIGUMS	67
INFORMĀCIJAS AVOTI	68
PIELIKUMI.....	69
1.pielikums. Programmas pirmteksts.....	69

IEVADS

Šobrīd ir ļoti svarīgi izmantot vienoto digitalizēto risinājumu, kas ļauj viegli pārvaldīt datus un procesus iekš kādas iestādes, tajā skaitā deju skolās, kolektīvos un teātros. Uz doto brīdi iepriekš minētajām organizācijām nav pieejama sistēma, kas minimizētu sarežģījumus tērpu izsniegšanas un savākšanas posmā, grupu komplektēšanas laikā, kā arī koncerta programmas sastādīšanas un pasākuma organizācijas periodā. Lai novērstu šīs problēmas un izvairītos no cilvēciskajām kļūdām informācijas apstrādes un uzglabāšanas laikā, ir nepieciešams izstrādāt WEB aplikāciju, kas ļaus darbiniekiem un audzēkņiem piekļūt viņiem piemērotas informācijas un funkcionalitātes klāstam. Piekļuve centralizētai informācijas datu bāzei veicinās efektīvāku sadarbību starp skolotājiem, audzēkņiem un vecākiem, kā arī uzlabos vispārējo organizācijas darba plūsmu.

Tirgus izpēti un teātriem, deju kolektīviem un sporta komandām pieejamo risinājumu analīzes laikā, tika secināts, ka lielam kolektīvam piemērota tērpu pārvaldes tehnoloģija šobrīd nav pieejama tirgū. Izpētot pieredzes stāstus un atsauksmes, ar kurām teātru direktori un kolektīvu vadītāji dalījās tīmeklī, tika atklāts, ka lielākā daļa ar skatuves mākslu saistīto institūciju seko līdzīgai savām kostīmu krātuvēm, izmantojot Excel tabulas vai papīra žurnālus, retos gadījumos tiek izmantota kāda inventāra pārvaldes sistēma. Šāda situācija norāda uz būtisku robu digitālajos risinājumos, kas varētu palīdzēt skatuves mākslas iestādēm efektīvāk pārvaldīt savus resursus. Tradicionālie uzskaites veidi bieži vien rada papildu administratīvo slogu un palielina kļūdu iespējamību, īpaši lielās organizācijās, kur ir daudz dalībnieku un inventāra vienību.

Viens no tādas sistēmas piemēriem, ko izmanto kādā no ASV teātriem, ir StageStock. Šī ir viena no visveiksmīgāk izveidotajām inventarizācijas sistēmām, kas šobrīd ir pieejama lietošanai. Mākoņu risinājums StageStock ļauj administratoram samērā viegli veikt kādas institūcijas īpašumā esošo tērpu, aksesuāru un rekvizītu uzskaiti, nodrošinot ērtu vienību pievienošanu, pārvaldīšanu un meklēšanu. Starp stiprajām pusēm ir jāpiemin arī iespēja pievienot jebkāda veida failus vienības aprakstam un apskatīt tās vēsturi. Šo funkcionalitāti ir mērķis ieviest arī projektā - skolas sistēmā. Vislielākais trūkums ir tas, ka pirms lietošanas sistēmas lietotne ir jālejupielādē un tā ir pieejama tikai veicot ikmēneša maksājumu. Turklāt, sistēmas lietošanas izmaksas var kļūt par būtisku šķērslī, īpaši mazām organizācijām ar ierobežotiem budžetiem, kas ne vienmēr var atļauties regulāras abonēšanas maksas.

Cits risinājums ar līdzīgu funkcionalitāti, ko forumā bija pieminējusi kāda Lielbritānijas deju kolektīva vadītāja, ir mobilā lietotne krātuvju pārvaldei StockControl. Šī risinājuma lielākās priekšrocības ir sekojošas: tas ir portatīvs un viegli lietojams, tas ļauj

sadalīt un sagrupēt vienības pēc kategorijas un atrašanās vietas, kā arī sistēma saglabā un apkopo statistikas datus. Sistēmas vājā puse ir tāda, ka StockControl lietotne ir pieejama tikai uz mobilajām ierīcēm ar iOS operētājsistēmu, kas traucē to izmantot lielai lietotāju grupai. Ierobežotā pieejamība un nepieciešamība pēc specifiska tehniskā aprīkojuma var būtiski samazināt tādas sistēmas efektivitāti.

Pēc konkurentu piedāvājumu analīzes veikšanas var secināt, ka neviena no šīm sistēmām nav pilnībā piemērota deju skolas vai teātra tērpu pārvaldīšanai, jo katra izstrādātāja mērķis ir izveidot produktu, kas būs piemērots pēc iespējas lielākai lietotāju grupai. Tas pozitīvi atspoguļojas uz konkrētā risinājuma biznesa aspektiem, bet negatīvi ietekmē lietotāju pieredzi darbā ar sistēmu. Vēl viens kopīgs trūkums, kas ir šīm sistēmām, ir tas, ka, lai tās lietotu, ir jālejupielādē mobilā vai darbvirsma aplikācija un jāsamaksā par tās lietošanu, ko neprofesionālām studijām un teātriem ne vienmēr ir iespēja izdarīt. Šo ierobežojumu dēļ daudzas organizācijas izvēlas palikt pie novecojušām metodēm, neskatoties uz to neefektivitāti un lielāku kļūdu risku.

Risinājums, kas tiks izstrādāts šī darba ietvaros, atšķirsies no konkurentu piedāvātajām sistēmām ar to, ka būs piemērots tieši vienas konkrētās deju skolas vajadzībām, būs pieejams bez maksas un nepieprasīs atsevišķas lietotnes ielādi. Tas atvieglos skolas direktora un pedagoga darbu, nesagādājot liekas tehniskās un finanšu grūtības.

1. UZDEVUMA NOSTĀDNE

Kvalifikācijas darba uzdevums ir izveidot baletskolas kostīmu uzskaites sistēmu. Sistēmā nepieciešams realizēt iespēju glabāt un pārvaldīt datus par skolas pedagogiem, audzēkņiem un viņu vecākiem, kā arī nodarbību grafikiem un kostīmiem. Pedagogiem un administrācijai ir jābūt iespējai ērti piekļūt sarakstiem ar saviem audzēkņiem, koncertiem, nodarbībām un noliktavā esošiem kostīmiem, audzēkņiem un vecākiem jābūt nodrošinātai piekļuvei informācijai par saviem pedagogiem, izpildāmiem numuriem un tiem nepieciešamiem tērpiem. Papildus iepriekš minētajām prasībām skolas direktoram ir jānodrošina iespēja pievienot un pārvaldīt informāciju par koncertiem, numuriem, tiem piesaistītiem audzēkņiem, pedagogiem un tērpiem. Skolas personālam ir jābūt iespējai veikt nepieciešamās informācijas izdrukas ērtai piekļuvei vajadzīgiem datiem.

Ir plānotas vairākas funkcijas (skat. 1.att.):



1.att. Lietojumgadījuma diagramma

- skolas direktoram (administratoram) būs iespēja:
 - pievienot, dzēst un rediģēt informāciju par skolas pedagogiem, audzēkņiem, viņu vecākiem un audzēkņu grupām;

- pievienot, dzēst un rediģēt informāciju par noliktavā esošiem un audzēkņiem izdalītiem kostīmiem;
- pievienot, dzēst un rediģēt informāciju par koncertiem;
- pievienot, dzēst un rediģēt informāciju par koncertos iekļautiem numuriem, numuru horeogrāfiem, izpildītājiem un nepieciešamiem kostīmiem;
- veikt sev piemērotas audzēkņu, pedagogu, grupu un tērpu sarakstu izdrukas;
- izdrukāt automātiski sastādītu koncerta numuru sarakstu;
- veikt jauno lietotāju kontu ar noteiktu lomu pievienošanu sistēmai;
- skolas pedagogam būs iespēja:
 - piekļūt informācijai par savu grafiku, grupām, to audzēkņiem, viņu vecākiem un kostīmiem;
 - piekļūt informācijai par koncertiem, tajos iekļautiem numuriem, izpildītājiem un nepieciešamiem kostīmiem;
 - veikt nepieciešamās savu grupu audzēkņu un viņu vecāku sarakstu izdrukas;
 - izdrukāt pedagoga audzēkņiem nepieciešamo tērpu sarakstus;
 - izdrukāt pedagoga studētu koncerta numuru sarakstu;
- skolas audzēknim būs iespēja:
 - piekļūt informācijai par savu grupu, nodarbību sarakstu, pedagogu un kostīmiem;
 - redzēt informāciju par koncertiem un numuriem, kuros audzēknis piedalās un tiem nepieciešamiem tērpiem;
- audzēkņu vecākiem būs iespēja:
 - piekļūt informācijai par sava bērna grupu, nodarbību sarakstu, pedagogu un kostīmiem;
 - redzēt informāciju par koncertiem un numuriem, kuros audzēkņi ir iesaistīti;
 - redzēt un izdrukāt informāciju par nepieciešamiem tērpiem.

2. PRASĪBU SPECIFIKĀCIJA

2.1. Ieejas un izejas informācijas apraksts

2.1.1. Ieejas informācijas apraksts

Sistēmā tiks nodrošināta šādas ieejas informācijas, ko ievadīs lietotājs no tastatūras apstrāde.

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

- Personas kods – lietotāja personas kods – burtu teksts ar izmēru 12 rakstzīmes(piem. 160906-27839).
- Vārds - lietotāja vārds - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. Alise).
- Uzvārds - lietotāja uzvārds - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. Siliņa).
- E-pasts - lietotāja elektroniskā pasta adrese - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. alisesilina@gmail.com).
- Parole – parole, ar kuru lietotājs autorizējās sistēmā - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. Silina!123).
- Tālrunis – lietotāja telefona numurs - burtu teksts ar izmēru 12 rakstzīmes (piem. +37198456763).
- Dzimšanas diena – lietotāja dzimšanas dienas datums – datums (piem. 16.09.2006.).

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

- Nosaukums – lomas nosaukums - burtu teksts ar izmēru līdz 30 rakstzīmēm(piem. Audzēknis).

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

- Nosaukums – kostīma nosaukums - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. svārki).
- Krāsa – kostīma krāsa - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. rozā).
- Izmērs – kostīma izmērs - burtu teksts ar izmēru līdz 20 rakstzīmēm (piem. 165 cm).
- Attēls – foto attēls, kas demonstrē kostīma izskatu – grafiskais fails

4. Informācija par **grupām** sastāvēs no šādiem datiem.

- Nosaukums – grupas nosaukums - burtu teksts ar izmēru līdz 5 rakstzīmēm(piem. P2).
- Grafiks – nodarbību grafiks – teksts (piem.pirmdiena, trešdiena 17:00-18:00)

5. Informācija par **filiālēm** sastāvēs no šādiem datiem.

- Valsts – filiāles atrašanas valsts - burtu teksts ar izmēru līdz 20 rakstzīmēm (piem. Latvija).
- Pilsēta – filiāles atrašanas pilsēta - burtu teksts ar izmēru līdz 20 rakstzīmēm (piem. Rīga).
- Iela – filiāles atrašanas iela - burtu teksts ar izmēru līdz 20 rakstzīmēm (piem. Bauskas iela).
- Ēka – filiāles atrašanas ēkas numurs - burtu teksts ar izmēru līdz 20 rakstzīmēm (piem. 34A).
- Indekss – filiāles atrašanas adreses indekss - burtu teksts ar izmēru 7 rakstzīmes (piem. LV-1014).

6. Informācija par **numuriem** sastāvēs no šādiem datiem.

- Nosaukums – numura nosaukums - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. Ziedu valsis).
- Kārtas numurs - priekšnesuma kārtas numurs koncertā - cipars (piem. 2).
- Garums – priekšnesuma ilgums - burtu teksts ar izmēru līdz 6 rakstzīmēm (piem. 2:30).
- Mūzika - mūzikas kompozīcijas nosaukums - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. Ziedu simfonija).
- Horeogrāfija - horeogrāfijas stila nosaukums - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. klasiskā deja).

7. Informācija par **koncertiem** sastāvēs no šādiem datiem.

- Nosaukums – koncerta nosaukums - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. Ziedu dārzi).
- Datums – koncerta norises datums – datums (piem. 2024-02-17)
- Vieta – koncerta norises vieta - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. VEF kultūras pils).
- Saite – saite uz biļešu tirgotāja mājaslapu - burtu teksts ar izmēru līdz 60 rakstzīmēm (piem. <https://www.bilesuserviss.lv/lat/ziedudarzi>).
- Ilgums – koncerta garums - burtu teksts ar izmēru līdz 10 rakstzīmēm (piem. 2 stundas).

2.1.2. Izejas informācijas apraksts

1. **PDF izdruka** ar grupas audzēkņu sarakstu, kur lapas augšējā daļā ir norādīts grupas nosaukums, nodarbību grafiks, pasniedzējs un audzēkņu sakits, zemāk jāatrodas audzēkņu saraksts tabulas formātā. Tabulā jābūt audzēkņa vārdam, uzvārdam un telefona numuram, kā arī audzēkņa vecāka vārdam, uzvārdam un telefona numuram.
2. **PDF izdruka** ar noliktavā pieejamu tērpu sarakstu, kur lapas augšējā daļā ir jābūt norādītam pieejamo tērpu skaitam, kam sekos tērpu saraksts tabulas formātā ar tērpu nosaukumiem, krāsām, izmēriem un attēliem.
3. **PDF izdruka** ar koncerta numuru sarakstu. Lapas augšējā daļā vidū ir jābūt virsrakstam ar koncerta nosaukumu, zem tā jābūt norādītam koncerta datumam, vietai, ilgumam un numuru skaitam. Dokumenta pamatdaļā jāatrodas numuru sarakstam, kur priekšnesumi ir attēloti atbilstoši to kārtas numuriem. Numuru sarakstam jābūt noformētam tabulas veidā un jāsaturs priekšnesuma kārtas numuru, nosaukumu, audzēkņa, kas izpildīs numuru vārdu un uzvārdu, garumu, horeogrāfijas veida nosaukumu un pedagoga, kas studēja numuru vārdu un uzvārdu.
4. **PDF izdruka** ar konkrētam koncertam nepieciešamo kostīmu sarakstu. Lapas augšējā daļā vidū ir jābūt virsrakstam ar koncerta nosaukumu, zem tā jābūt norādītam koncerta datumam, vietai, ilgumam un kopējam izpildītāju skaitam. Dokumenta pamatdaļā tabulas veidā jābūt attēlotam kostīmu sarakstam, kas sastāvēs no numura nosaukuma, izpildītāju skaita, kostīma nosaukuma un kostīma attēla.
5. **PDF izdruka** ar pedagoga audzēkņiem nepieciešamo tērpu sarakstu, kur lapas augšējā daļā vidū ir jābūt virsrakstam ar pedagoga vārdu un koncerta nosaukumu, zem tā jābūt norādītam koncerta datumam, vietai un ilgumam. Dokumenta pamatdaļā tabulas veidā jābūt attēlotam kostīmu sarakstam, kas sastāvēs no numura nosaukuma, izpildītāju skaita, kostīma nosaukuma un kostīma attēla.
6. **PDF izdruka** ar koncerta numuriem, ko studēja konkrēts pedagogs. Lapas augšējā daļā vidū ir jābūt virsrakstam ar pedagoga vārdu un koncerta nosaukumu, zem tā jābūt norādītam koncerta datumam, vietai un ilgumam. Dokumenta pamatdaļā jāatrodas numuru sarakstam, kur priekšnesumi ir attēloti atbilstoši to kārtas numuriem. Numuru sarakstam jābūt noformētam tabulas veidā un jāsaturs priekšnesuma kārtas numuru, nosaukumu, audzēkņa, kas izpildīs numuru vārdu un uzvārdu, garumu un horeogrāfijas veida nosaukumu.
7. **PDF izdruka** vecākam, kur viņš var redzēt tērpus, kas būs nepieciešami viņa bērnam konkrētajā koncertā. Lapas augšējā daļā vidū ir jābūt virsrakstam ar koncerta

nosaukumu un bērnu vārdu un uzvārdu, zem tā jābūt norādītam koncerta datumam, vietai un ilgumam. Dokumenta pamatdaļā tabulas veidā jābūt attēlotam kostīmu sarakstam, kas sastāvēs no numura nosaukuma, kostīma nosaukuma un kostīma attēla.

2.2. Funkcionālās prasības

1. Jauno lietotāju kontu pievienošana:
 - 1.1. Jānodrošina, ka jaunu lietotāja kontu var pievienot tikai lietotājs ar administratora lomu.
 - 1.2. Jāļauj izveidot jaunu lietotāju kontu, izmantojot formu un norādot personas datus un izvēloties lietotāja lomu.
 - 1.3. Ja obligātie lauki nav aizpildīti, neļaut izveidot kontu un izvadīt ziņojumu.
 - 1.4. Ja formāts neatbilst ieejas informācijas apraksta 1. punktā minētajam, neļaut pievienot kontu un izvadīt ziņojumu.
 - 1.5. Ja informācija par lietotāju ar ievadīto personas kodu jau pastāv sistēmā, neļaut pievienot kontu un izvadīt ziņojumu.
2. Lietotāja autorizācija:
 - 2.1. Lietotājiem jānodrošina autorizācija sistēmā ar kontu, ko viņiem izveidoja administrators, izmantojot formu, kur ievadīt personas kodu un paroli.
 - 2.2. Ja personas kods sakrīt ar datu bāzē esošā lietotāja personas kodu un parole ir atbilstoša šim lietotājam, novirzīt lietotāju uz viņa konta galveno izvēlni.
 - 2.3. Ja personas kods vai parole nesakrīt ar sistēmā esošo, izvadīt ziņojumu un neļaut piekļūt kontam.
3. Lietotāju kontu rediģēšana un dzēšana:
 - 3.1. Jānodrošina, ka lietotājs, kuram ir pedagoga vai audzētāja tiesības var tikai rediģēt savu profilu.
 - 3.2. Lietotājs ar vecāka tiesībām var tikai rediģēt savu vai sava bērna profilu.
 - 3.3. Rediģēt un dzēst informāciju par jebkuru lietotāju var tikai administrators.
 - 3.4. Lietotāju profilu rediģēšanai ir jāpiedāvā forma, kur jāievada jauni dati.
 - 3.5. Ja formāts neatbilst ieejas informācijas apraksta 1. punktā minētajam, neļaut pievienot informāciju un izvadīt ziņojumu.
 - 3.6. Pirms jaunas informācijas profila dzēšanas jāizvada brīdinājuma ziņojums ar iespēju saglabāt vai atcelt veiktās izmaiņas.
4. Informācijas par grupām pārvaldīšana:

- 4.1. Iespēja pārvaldīt informāciju par grupām ir jānodrošina tikai lietotājiem ar administrātoru lom.
 - 4.2. Jāļauj izveidot jaunu grupas ierakstu, izmantojot formu un norādot grupas nosaukumu, pedagogu un filiāli.
 - 4.3. Ja obligātie lauki nav aizpildīti, neļaut izveidot ierakstu un izvadīt ziņojumu.
 - 4.4. Ja formāts neatbilst ieejas informācijas apraksta 4. punktā minētajam, neļaut pievienot grupu un izvadīt ziņojumu.
 - 4.5. Grupu rediģēšanai ir jāpiedāvā forma, kur jāievada jauni dati paratbilstošo grupu.
 - 4.6. Ja jauno datu formāts neatbilst ieejas informācijas apraksta 4. punktā minētajam, neļaut rediģēt informāciju par grupu un izvadīt ziņojumu.
 - 4.7. Pirms grupas dzēšanas jāizvada brīdinājuma ziņojums ar iespēju saglabāt vai atcelt veiktās izmaiņas
5. Informācijas par kostīmiem pārvaldīšana:
 - 5.1. Iespēja pārvaldīt informāciju par kostīmiem ir jānodrošina tikai lietotājiem ar administrātoru lom.
 - 5.2. Jāļauj izveidot jaunu kostīma ierakstu, izmantojot formu un norādot kostīma nosaukumu, krāsu, veidu, izmēru un pievienojot attēlu.
 - 5.3. Ja obligātie lauki nav aizpildīti, neļaut izveidot ierakstu un izvadīt ziņojumu.
 - 5.4. Ja formāts neatbilst ieejas informācijas apraksta 3. punktā minētajam, neļaut pievienot tērpu un izvadīt ziņojumu.
 - 5.5. Tērpu rediģēšanai ir jāpiedāvā forma, kur jāievada jauni dati paratbilstošo tērpu.
 - 5.6. Ja jauno datu formāts neatbilst ieejas informācijas apraksta 3. punktā minētajam, neļaut rediģēt informāciju par kostīmu un izvadīt ziņojumu.
 - 5.7. Pirms informācijas rediģēšanas vai kostīma dzēšanas jāizvada brīdinājuma ziņojums ar iespēju saglabāt vai atcelt veiktās izmaiņas
 6. Kostīmu izdalīšana un savākšana:
 - 6.1. Kostīmu izdalīšanas un savākšanas funkcionalitāte ir jānodrošina tikai lietotājiem ar administrātoru vai pedagoga tiesībām.
 - 6.2. Pēc kostīma izvēles jāpiedāvā audzēkņu saraksts, kuriem ir iespēja izdalīt kostīmu.
 - 6.3. Lai savāktu kostīmu, pēc kostīma izvēles no lietotāju saraksta ir jāizvēlas persona, kura kostīmu ir atgriezusi.
 7. Piekļuve informācijai par lietotājiem:
 - 7.1. Piekļuve informācijai par lietotājiem ir jānodrošina tikai personām ar administrātoru, pedagoga vai vecāka tiesībām.

- 7.2. Lietotājam ar pedagoga lomu jānodrošina piekļuve tikai viņam piesaistītā grupā esošiem lietotājiem.
- 7.3. Lietotājam ar vecāka lomu jānodrošina piekļuve tikai viņam piesaistīto bērnu informācijai.
- 7.4. Lietotājiem ar atbilstošām tiesībām ir jāspēj filtrēt viņiem pieejamo informāciju pēc grupas.
- 7.5. Lietotājiem ar atbilstošām tiesībām ir jāspēj veikt meklēšanu pieejamā informācijas klāstā.
8. Piekļuve informācijai par kostīmiem:
 - 8.1. Piekļuve informācijai par visiem kostīmiem ir jānodrošina tikai lietotājiem ar administratora vai pedagoga lomu.
 - 8.2. Lietotājiem ar bērna vai vecāka lomu jānodrošinā piekļuve informācijai tikai par bērnam izdalītiem kostīmiem.
 - 8.3. Ir nepieciešams nodrošināt iespēju veikt meklēšanu pēc atslēgas vārdiem.
9. Informācijas par numuriem pārvaldīšana:
 - 9.1. Iespēja pārvaldīt informāciju par numuriem ir jānodrošina tikai lietotājiem ar administrātoru lomu.
 - 9.2. Jāļauj izveidot jaunu numura ierakstu, izmantojot formu un norādot numura nosaukumu, kārtas numuru, garumu, mūziku, horeogrāfijas stilu un pievienojot izpildītājus.
 - 9.3. Ja obligātie lauki nav aizpildīti, neļaut izveidot ierakstu un izvadīt ziņojumu.
 - 9.4. Ja formāts neatbilst ieejas informācijas apraksta 6. punktā minētajam, neļaut pievienot numuru un izvadīt ziņojumu.
 - 9.5. Numuru rediģēšanai ir jāpieāvā forma, kur jāievada jauni dati par atbilstošo numuru.
 - 9.6. Ja jauno datu formāts neatbilst ieejas informācijas apraksta 6. punktā minētajam, neļaut rediģēt informāciju par numuru un izvadīt ziņojumu.
 - 9.7. Pirms informācijas rediģēšanas vai numura dzēšanas jāizvada brīdinājuma ziņojums ar iespēju saglabāt vai atcelt veiktās izmaiņas.
10. Informācijas par koncertiem pārvaldīšana:
 - 10.1. Iespēja pārvaldīt informāciju par koncertiem ir jānodrošina tikai lietotājiem ar administrātoru lomu.
 - 10.2. Jāļauj izveidot jaunu koncerta ierakstu, izmantojot formu un norādot koncerta nosaukumu, datumu, vietu, ilgumu, saiti uz biļešu tirgotāju un pievienojot numurus.
 - 10.3. Ja obligātie lauki nav aizpildīti, neļaut izveidot ierakstu un izvadīt ziņojumu.

- 10.4. Ja formāts neatbilst ieejas informācijas apraksta 7. punktā minētajam, neļaut pievienot koncertu un izvadīt ziņojumu.
- 10.5. Koncertu rediģēšanai ir jāpiedāvā forma, kur jāievada jauni dati par atbilstošo koncertu.
- 10.6. Ja jauno datu formāts neatbilst ieejas informācijas apraksta 7. punktā minētajam, neļaut rediģēt informāciju par koncertu un izvadīt ziņojumu.
- 10.7. Pirms informācijas rediģēšanas vai koncerta dzēšanas jāizvada brīdinājuma ziņojums ar iespēju saglabāt vai atcelt veiktās izmaiņas.
11. Grupas Audzēkņu Saraksta Izdruka:
 - 11.1. Lietotājam ar administratora tiesībām jābūt iespējai izvēlēties grupu no pieejamo grupu saraksta.
 - 11.2. Pēc grupas izvēles, sistēmai jāizvadī dati par grupu, tostarp grupas nosaukums, nodarbību grafiks, pasniedzējs un audzēkņu skaits.
 - 11.3. Sistēmai jāģenerē PDF dokuments, kurā ietverts audzēkņu saraksts ar vārdiem, uzvārdiem, telefona numuriem, kā arī audzēkņu vecāku vārdiem, uzvārdiem un telefona numuriem.
 - 11.4. PDF dokumentu jābūt iespējai lejupielādēt un izdrukāt.
12. Noliklavā Pieejamo Tēru Saraksta Izdruka:
 - 12.1. Lietotājam ar administratora tiesībām jābūt iespējai piekļūt noliklavā pieejamo tēru sarakstam.
 - 12.2. Sistēmai jāizvadī dati par pieejamo tēru skaitu, tostarp tēru nosaukumi, krāsas, izmēri un attēli.
 - 12.3. Sistēmai jāģenerē PDF dokuments, kurā ietverts pieejamo tēru saraksts.
 - 12.4. PDF dokumentu jābūt iespējai lejupielādēt un izdrukāt.
13. Koncerta Numuru Saraksta Izdruka:
 - 13.1. Lietotājam ar administratora tiesībām jābūt iespējai piekļūt koncerta numuru sarakstam, izvēloties konkrētu koncertu.
 - 13.2. Pēc koncerta izvēles, sistēmai jāizvadī koncerta nosaukums, datums, vieta, ilgums un numuru skaits.
 - 13.3. Sistēmai jāģenerē PDF dokuments, kurā ietverts numuru saraksts ar priekšnesuma kārtas numuru, nosaukumu, audzēkņa vārdu un uzvārdu, garumu, horeogrāfijas veidu un pedagoga vārdu un uzvārdu.
 - 13.4. PDF dokumentu jābūt iespējai lejupielādēt un izdrukāt.
14. Koncertam Nepieciešamo kostīmu Saraksta Izdruka:

- 14.1. Lietotājam ar administrātoru tiesībām jābūt iespējai piekļūt koncertam nepieciešamo
kostīmu sarakstam, izvēloties konkrētu koncertu.
- 14.2. Pēc koncerta izvēles, sistēmai jāizvadī koncerta nosaukums, datums, vieta, ilgums
un kopējais izpildītāju skaits.
- 14.3. Sistēmai jāģenerē PDF dokuments, kurā ietverts kostīmu saraksts ar numura
nosaukumu, izpildītāju skaitu, kostīma nosaukumu un kostīma attēlu.
- 14.4. PDF dokumentu jābūt iespējai lejupielādēt un izdrukāt.
15. Pedagoģa audzēkņiem nepieciešamo tērpu saraksta Izdruka:
 - 15.1. Pedagoģam jābūt iespējai piekļūt audzēkņiem nepieciešamo tērpu sarakstam,
izvēloties konkrētu koncertu un savu grupu.
 - 15.2. Pēc koncerta izvēles, sistēmai jāizvadī koncerta nosaukums, datums, vieta un
ilgums.
 - 15.3. Sistēmai jāģenerē PDF dokuments, kurā ietverts kostīmu saraksts ar numura
nosaukumu, izpildītāju skaitu, kostīma nosaukumu un kostīma attēlu.
 - 15.4. PDF dokumentu jābūt iespējai lejupielādēt un izdrukāt.
16. Koncerta numuru iaraksta Izdruka pedagoģam:
 - 16.1. Pedagoģam jābūt iespējai piekļūt koncerta numuru sarakstam, kuros viņš piedalās,
izvēloties konkrētu koncertu.
 - 16.2. Pēc koncerta izvēles, sistēmai jāizvadī koncerta nosaukums, datums, vieta un
ilgums.
 - 16.3. Sistēmai jāģenerē PDF dokuments, kurā ietverts numuru saraksts ar priekšnesuma
kārtas numuru, nosaukumu, audzēkņa vārdu un uzvārdu, garumu un horeogrāfijas
veidu.
 - 16.4. PDF dokumentu jābūt iespējai lejupielādēt un izdrukāt.
17. Tērpu Saraksta Izdruka Vecākiem:
 - 17.1. Audzēkņa vecākiem jābūt iespējai piekļūt tērpu sarakstam, kas būs nepieciešami
viņu bērnam konkrētajā koncertā.
 - 17.2. Pēc koncerta izvēles, sistēmai jāizvadī koncerta nosaukums, bērna vārds un uzvārds,
datums, vieta un ilgums.
 - 17.3. Sistēmai jāģenerē PDF dokuments, kurā ietverts kostīmu saraksts ar numura
nosaukumu, kostīma nosaukumu un kostīma attēlu.
 - 17.4. PDF dokumentu jābūt iespējai lejupielādēt.
18. Piekļuve informācijai par numuriem:

- 18.1. Lietotājiem ar administratora vai pedagoga lomu jābūt iespējai piekļūt informācijai par visiem numuriem.
- 18.2. Pedagogiem jābūt iespējai piekļūt informācijai tikai par tiem numuriem, kurus viņi studē.
- 18.3. Audzēkņiem un vecākiem jābūt iespējai piekļūt informācijai tikai par tiem numuriem, kuros audzēknis piedalās.
- 18.4. Informācijai jābūt pieejamai tabulas.
19. Piekļuve informācijai par koncertiem:
 - 19.1. Lietotājiem ar administratora lomu jābūt iespējai piekļūt informācijai par visiem koncertiem.
 - 19.2. Pedagogam jābūt iespējai piekļūt informācijai tikai par tiem koncertiem, kuros tiek izpildīti viņa numuri.
 - 19.3. Audzēkņiem un vecākiem jābūt iespējai piekļūt informācijai tikai par tiem koncertiem, kuros audzēknis piedalās.
 - 19.4. Informācijai jābūt pieejamai tabulas veidā

2.3. Nefunkcionālās prasības

1. Sistēmas saskarnes valodai ir jābūt latviešu valodai.
2. Jānodrošina tas, ka tīmekļa lietojumprogramma automātiski pielāgojas un saglabā ērtu lietojamību dažādos ekrāna izmēros.
3. Sistēma ir jābūt saderīgai ar populārākajiem tīmekļa pārlūkiem, piemēram, Chrome, Firefox, un Safari.
4. Dizainam ir jābūt zilos toņos.
5. Sistēmai jāsniedz lietotājam informatīvus paziņojumus par veiksmīgiem vai neveiksmīgiem procesiem, nodrošinot skaidru lietojamību.

6. Lietotāja pieslēgšanās formai jāizskatās sekojoši (skat. 2.att).



A sketch of a user login form. It is a rounded rectangle with a light gray border. At the top, the title "Pieslēgšanās" is centered in a bold, black, sans-serif font. Below the title, there are two input fields, each with a light gray border and rounded ends. The first field contains the placeholder text "personas kods" in a light gray font. The second field contains the placeholder text "parole" in a light gray font. Below these two fields, there is a single button with a light gray border, rounded ends, and the text "Ienākt" in a bold, black, sans-serif font.

2.att. Lietotāja pieslēgšanas formas skice

7. Lietotāja konta rediģēšanas formai jāizskatās sekojoši (skat. 3.att).



A sketch of a user account editing form. It is a rounded rectangle with a light gray border. At the top, the title "Rediģēt manu kontu" is centered in a bold, black, sans-serif font. Below the title, there are four input fields, each with a light gray border and rounded ends. The first field contains the placeholder text "Vārds" in a light gray font. The second field contains the placeholder text "Uzvārds" in a light gray font. The third field contains the placeholder text "personas kods" in a light gray font. The fourth field contains the placeholder text "parole" in a light gray font. Below these four fields, there is a single button with a light gray border, rounded ends, and the text "Rediģēt" in a bold, black, sans-serif font.

3.att. Lietotāja datu rediģēšanas formas skice

8. Brīdinājuma ziņojumam ir jāsiszskatās sekojoši (skat. 4.att.).

Vai vēlaties saglabāt izmaiņas?

Saglabāt

Atcelt

4.att. Brīdinājuma ziņojuma skice

9. Koncerta numuru sarakstam ir jāsiszskatās sekojoši (skat. 5.att.).

Grande Ballet Studio

Ziedu pasaka

Datums: 2024-02-17

Vieta: VEF kultūras pils


Ilgums: 2 stundas

Numuru skaits: 6

Kārtas numurs	Nosaukums	Izpildītāji	Garums	Horcoregrāfija	Pedagogi
1	Ziedu valsis	Eva Kalniņa (P4) Alise Silina (P2)	2:30	klasiskā deja	Iļze Krauze
2	Pavasara ziedi	Iļze Vētra (P2) Inese Ābele (P4) Maja Liepa (D1) Elza Ozoliņa (P4)	3:10	moderna deja	Iļze Krauze
3	Sāpju pasākis	Rūta Gulbe (P2) Līga Dreimanis (P5) Eva Kalniņa (P4) Vilma Kalniņa (P3) Elza Ozoliņa (P4)	3:00	contemporary	Iļze Krauze
4	Rudens lapa	Vilma Kalniņa (P3) Dace Birzniece (P5)	3:15	jazz	Māris Jansons
5	Sauļas deja	Elza Ozoliņa (P4) Līga Kalniņa (P1)	3:00	moderna deja	Edgars Grīns
6	Kalmu ezers	Jānis Dērlis (P4) Rūdolfs Liepiņš (P4) Raitis Zarins (P4)	3:05	folk	Edgars Grīns

5.att. Koncerta numuru saraksta skice

10. Grupas audzēkņu sarakstam ir jāsīzskatās sekojoši (skat. 6.att.).

 Grande Ballet Studio

S2

Grafiks: Pirmdienā, Trešdienā 17:00-18:00
 Pasniedzējs: Edgars Grīns
 Audzēkņu skaits: 4

Vārds	Uzvārds	Tālruna numurs	Vecāks (Vārds, Uzvārds, Tālrunis)
Alise	Siliņa	+37198446634	Inga Bērziņa (+37198457686)
Jānis	Bērziņš	+37188455645	Andris Kalniņš (+37198456769)
Līga	Kalniņa	+37188455765	Sandra Ozola (+37198456770)
Pēteris	Ozoliņš	+37198456677	Guna Kļava (+37198456771)

6.att. Grupas audzēkņu saraksta skice

3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS

Baletscolas kostīmu uzskaites sistēma ir web aplikācija, kas paredzēta dalībnieku, kostīmu un koncertu datu uzskaitēi baletscolas ietvaros. Tā sastāv no divām galvenajām daļām: lietotāja daļas un servera daļas.

Lietotāja daļas izstrādē tiks pielietotas trīs galvenās tehnoloģijas: **HTML5**, kas tiek izmantota modernās un dinamiskās tīmekļa lapas struktūras nodrošināšanai, **CSS Bootstrap 3.4**, kas ļauj viegli pielāgot izskatu un izkārtojumu, piedāvājot responsīvu dizainu un saglabājot konsistentu izskatu visās ierīcēs, **DomPdf**, kas ir PHP bibliotēka, kura ļauj ģenerēt PDF dokumentus no HTML un CSS koda, nodrošinot vienkāršu un efektīvu veidu, kā izveidot drukājamās dokumentācijas tieši no tīmekļa lapām, kā arī **JavaScript**, kas nodrošina interaktivitāti un ērtu ziņojumu sūtīšanu un saņemšanu, ietekmējot sistēmas dinamiku, kopumā šīs tehnoloģijas ļauj veidot patīkamu lietotāju pieredzi.

Servera puses izstrādē, savukārt, tiks izmantotas četras sekojošās tehnoloģijas: **PHP8.1** programmēšanas valodas jaunākā versija, kas izmanto jaunākās tehnoloģijas, lai veiksmīgi un droši apstrādātu datus servera pusē, PHP ietvars **Laravel 10**, kas ļauj ātri izstrādāt dinamiskās tīmekļa lietotnes, izmantojot iebūvētās funkcijas, kā arī **MySQL**, kas tiek izmantots kā datubāzes pārvaldības sistēma, nodrošinot efektīvu un drošu datu uzglabāšanu, kopā minētās tehnoloģijas nodrošina modernu lietotāja pieredzi un efektīvu datu pārvaldību.

Sistēmas izstrādes procesā tiks izmantotas divas **izstrādes vides**: **Visual Studio Code**, kas ir atvērtā koda izstrādes vide, paredzēta tīmekļa lietojumprogrammu izveidei, tā nodrošina daudzas funkcijas un papildinājumus, kas atvieglo koda rakstīšanu, atklāšanu un projektu pārvaldību, kā arī **Laragon**, kas ir lokāla izstrādes vide MySQL un PHP, tā ļauj ātri iestatīt lokālu vidi izstrādei un testēšanai, pirms sistēma tiek publicēta tiešsaistē, šī vide nodrošina vienkāršu konfigurāciju un drošu darbību, padarot izstrādes procesu ērtu un efektīvu.

4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

4.1. Sistēmas struktūras modelis

4.1.1. Sistēmas arhitektūra

Sistēmā tiks veidota no četrām apakšsistēmām (skat. 7.att): **kostīmu datu apstrādes** apakšsistēmas, **lietotāju datu apstrādes** apakšsistēmas, **grupu datu apstrādes** apakšsistēmas un **koncertu datu apstrādes** apakšsistēmas.

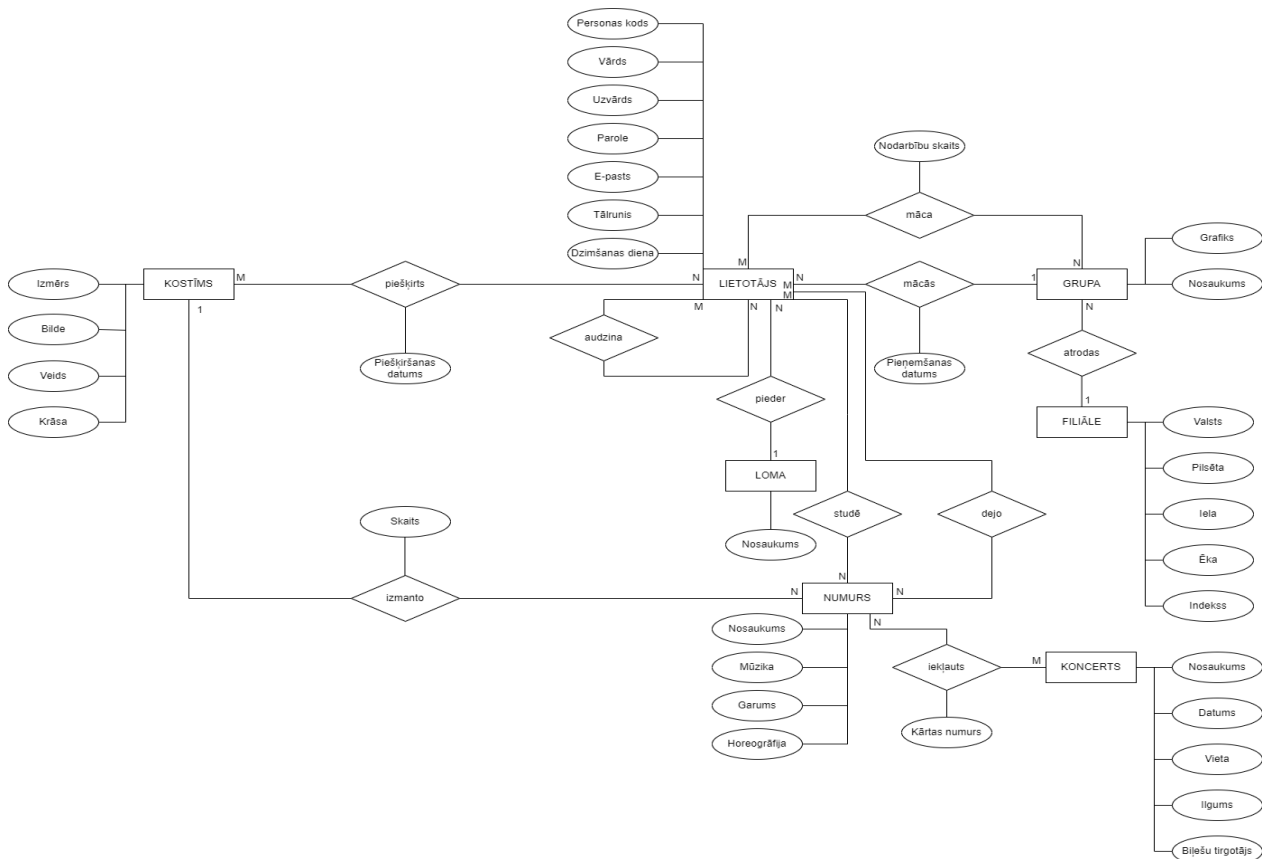


7.att. Funkcionālās dekompozīcijas diagramma

Kostīmu datu apstrādes apakšsistēmas astoņi moduļi nodrošinās lietotājam kostīmu pārvaldībai nepieciešamo funkcionalitāti, **lietotāju datu apstrādes apakšsistēmas** astoņi moduļi ļaus administratoram pārvaldīt lietotāju datus, kā arī lietotājiem veikt izmaiņas savā kontā, **grupu datu apstrādes apakšsistēma** sastāvēs no 12 moduļiem, nepieciešamiem lietotāju sadalīšanai pa grupām un darbam ar to datiem, **koncertu datu apstrādes apakšsistēmas** sastāv no četrpadsmit moduļiem un nodrošina koncertu un numuru datu pārvaldību un koncerta numuru un nepieciešamo kostīmu sarakstu izdruku veikšanu.

4.1.2. Sistēmas ER modelis

Sistēmas ER-modelis sastāv no 7 entitijām (skat. 8.att.), kas nodrošina pamat informācijas uzglabāšanu un apstrādi. Tie ir “LIETOTĀJS”, kas satur informāciju par sistēmas lietotājiem. Tās atribūtu kopums ietver sevī lietotāja personas kodu, vārdu, uzvārdu, elektronisko pastu, paroli, tālruni un dzimšanas dienu, “LOMA”, kas satur sevī informāciju par sistēmas lietotāju lomām, tās atribūtu kopums ietver lomas nosaukumu (piem. “audzēknis”, “administrators”), “KOSTĪMS”, kas satur informāciju par visiem kostīmiem, tās atribūtu kopums sevī ietver izmēru, veidu (piem. “svārki”, “kleita”), krāsu un kostīma bildi, “GRUPA”, kas satur sevī informāciju par audzēkņu grupām, tās atribūtu kopums sevī ietver nodarbību grafiku un grupas nosaukumu, “FILIĀLE”, kas satur sevī informāciju par filiāles atrašanās vietu, tās atribūtu kopums sevī ietver atrašanās valsti, pilsētu, ielu, ēku un tai atbilstošu pasta indeksu, “KONCERTS”, kas satur sevī informāciju par koncertu, tās atribūtu kopums ietver sevī koncerta nosaukumu, norises datumu, vietu, ilgumu un saiti uz biļešu tirgotāju “NUMURS”, kas satur sevī informāciju par numuru, kas tiek izpildīts koncertā, tās atribūtu kopums sevī ietver numura nosaukumu, garumu, mūzikas nosaukumu, izpildītāju skaitu un horeogrāfijas stila nosaukumu (klasiskā deja, laikmetīgā deja, utt.).



8.att. Sistēmas ER-diagramma

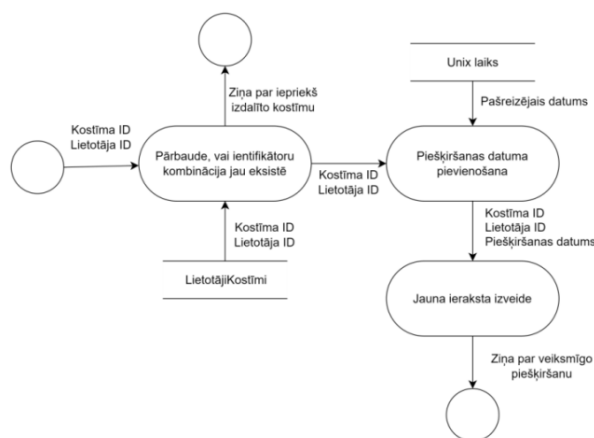
Lietotājus un lomas saista bināra saite ar kardinalitāti “viens pret daudziem”, jo vienam lietotājam var piederēt tikai viena loma, bet viena loma var piederēt vairākiem

lietotājiem, starp lietotājiem un kostītiem pastāv bināra saite ar kardinalitāti “daudzi pret daudziem”, jo viens kostīma veids var tikt piešķirts vairākiem lietotājiem kā arī vienam lietotājam var tikt piešķirti vairāki kostīmu veidi, lietotāju entitijas instances tika saistītas ar unāro relāciju saiti, jo lietotājam ar vecāka lomu tiek piesaistīti lietotāji ar audzēkņa lomu (viņu bērni), starp lietotājiem un grupām pastāv bināra saite ar kardinalitāti “daudzi pret daudziem”, jo viens lietotājs ar pedagoga lomu var mācīt vairākas grupas, kā arī vienai grupai var mācīt vairāki pedagogi, lietotājus un grupas saista arī bināra saite ar kardinalitāti “viens pret daudziem”, jo viens lietotājs ar lomu audzēknis var mācīties tikai vienā grupā, bet vienā grupā var mācīties vairāki audzēkņi, starp grupu un filiāli pastāv bināra saite ar kardinalitāti “viens pret daudziem”, jo grupa var atrasties tikai vienā filiālē, bet filiālē var būt vairākas grupas, starp lietotāju un numuru pastāv bināra saite ar kardinalitāti “daudzi pret daudziem”, jo viens lietotājs ar audzēkņa lomu var piedalīties vairākos numuros un vienā numurā var piedalīties vairāki lietotāji ar lomu audzēknis, kā arī vienu numuru var studēt vairāki lietotāji ar pedagoga lomu un viens lietotājs ar pedagoga lomu var studēt vairākus numurus, starp kostīmu un numuru pastāv bināra saite ar kardinalitāti “viens pret daudziem”, jo viens kostīma veids var tikt izmantots vairākos numuros, bet vienā numurā var būt izmantots tikai viens kostīma veids, starp numuru un koncertu pastāv bināra saite ar kardinalitāti “daudzi pret daudziem” jo vienā koncertā var tikt izpildīti vairāki numuri, kā arī viens numurs var tikt izpildīts vairākos koncertos.

4.2. Funkcionālais sistēmas modelis

4.2.1. Datu plūsmu modelis

Kostīma piešķiršana lietotājam (skat. 9.att.).

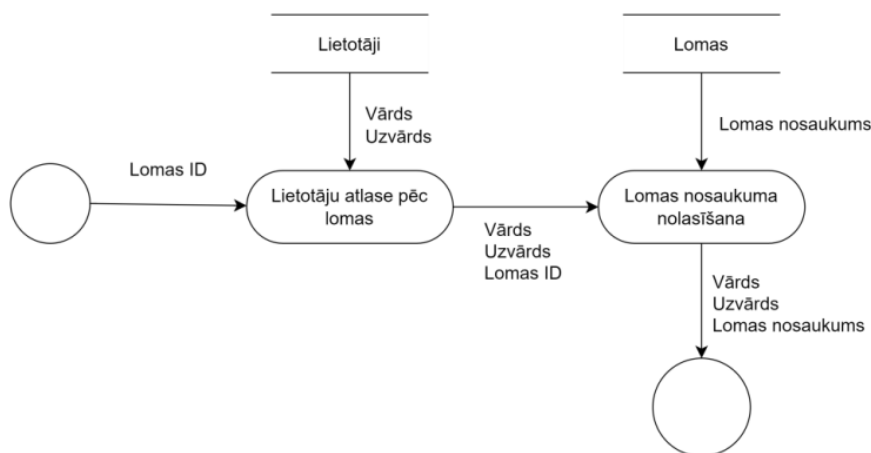


9.att. Kostīma piešķiršanas lietotājam datu plūsmu diagramma

Saskarsne padod kostīma vienības identifikatoru un izvēlēta lietotāja identifikatoru, ja kostīmu un lietotāju starptabulā nav ieraksta ar padotiem identifikātoriem, tad sistēma nolasa

tekošo datumu unix timestamp formātā un datubāzē izveido jaunu ierakstu kostīmu un lietotāju starptabulā ar kostīma un lietotāja identifikatoriem un kostīma piešķiršanas datumu, tiek parādīts veiksmes ziņojums, ja tāds ieraksts jau eksistē, tiek izvadīts informatīvais ziņojums par to, ka kostīms jau tika piešķirts lietotājam.

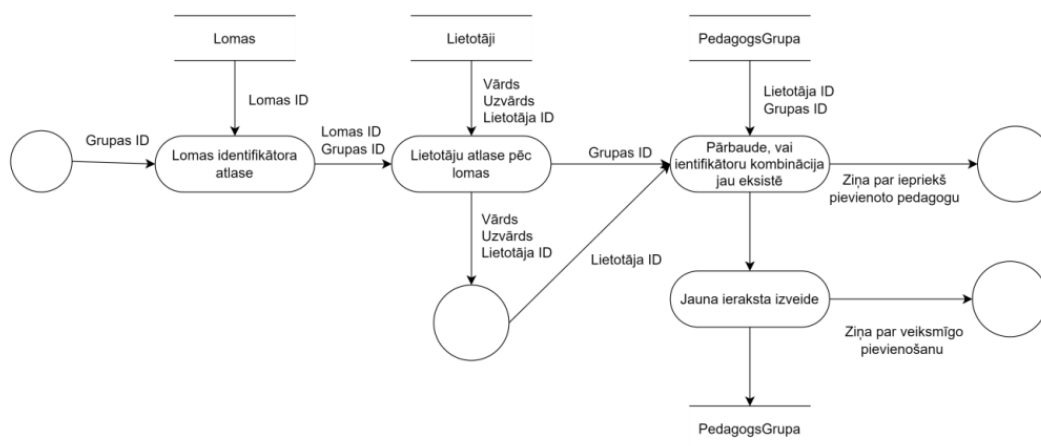
Lietotāja filtrēšana pēc lomas (skat. 10.att.).



10.att. Lietotāja filtrēšanas pēc lomas datu plūsmu diagramma

Saskarsne padod lomas identifikatoru, pēc kura no datubāzes lietotāju tabulas tiek atlasīti ieraksti ar atbilstošu lomu, no lomas tabulas tiek nolasīts lomas nosaukums, tad tiek attēlots lietotāja vārds, uzvārds un loma.

Pedagoga pievienošana grupai (skat. 11.att.).

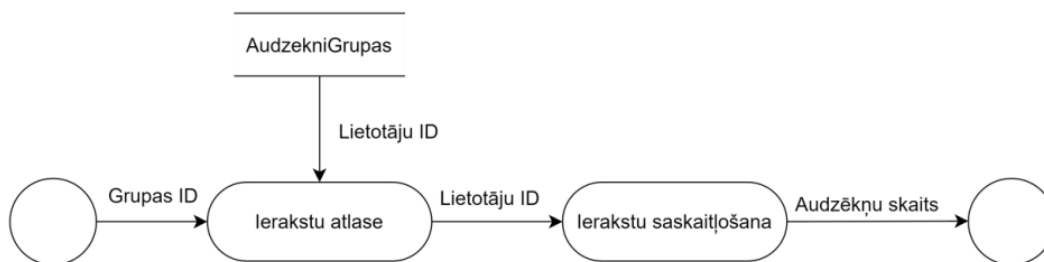


11.att. Pedagoga pievienošana grupai datu plūsmu diagramma

Saskarsne padod grupas identifikatoru, no lomu tabulas tiek atlasīts pedagoga lomas identifikators, no lietotāju tabulas tiek atlasīti lietotāji ar pedagoga lomas identifikatoru identifikatori, vārdi un uzvārdi un izvadīti saskarsnē, no saskarsnes tiek nolasīts lietotāja ID, notiek pārbaude, ja grupu un pedagogu starptabulā nav ieraksta ar padoto identifikatoru kombināciju, tabulai tiek pievienots jauns ieraksts un tiek izvadīts veiksmes ziņojums, ja

identifikatoru kombinācija dublējās, tiek izvadīts informatīvais ziņojums, ka pedagogs jau ir pievienots grupai.

Audzēkņu skaita noteikšana grupai (skat. 12.att.).

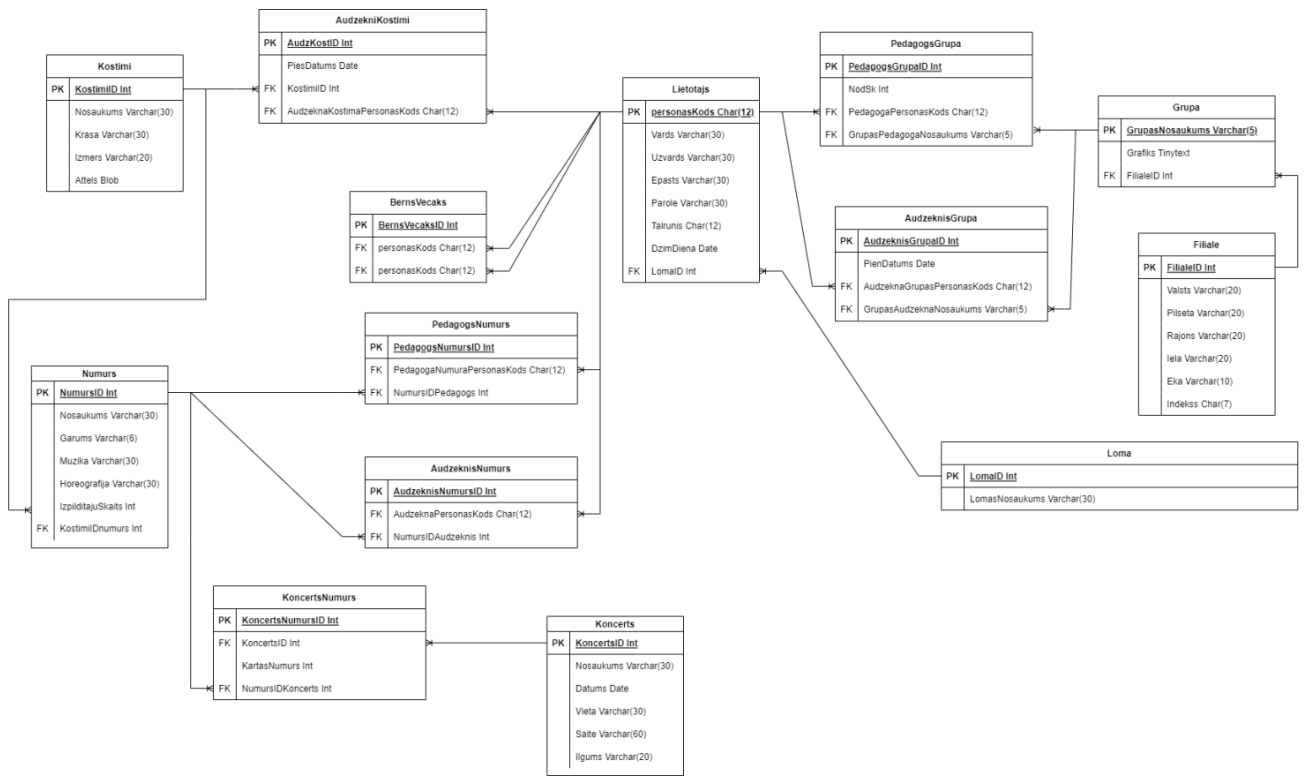


12.att. Audzēkņu skaita noteikšana grupai datu plūsmu diagramma

Saskarsne nolasa grupas identifikatoru, no audzēkņu un grupas starptabulas tiek atlasīti ieraksti ar atbilstošo grupas identifikatoru, tiek saskaitīti audzēkņu ID ierakstos ar atbilstošo grupas identifikatoru, rezultāts tiek izvadīts saskarsnē.

5. DATU STRUKTŪRU APRAKSTS

Datubāzes projektēšanas rezultātā tika izveidotas vairākas tabulas, un tika definēta to relācija jeb saistība (13. Att.).



13.att. Datubāzes tabulu shēma

Datubāze sastāv no 14 tabulām, kas satur informāciju par lietotājiem, viņu lomām, piederošiem kostīmiem, apmācāmām vai atbilstošām grupām, pašām grupām kā arī to atrašanās vietām, koncertiem, tajos iesaistītiem numuriem, numuru horeogrāfiem un izpildītājiem kā arī nepieciešamiem kostīmiem:

- tabula **“lietotajs”** sastāv no 8 laukiem un ir savienota ar tabulām “vestule” un “loma” ar saiti viens-pret-daudziem, “kostimi” un “grupa” ar saiti daudz-pret-daudziem un pati ar sevi ar unāro relācijas saiti,
- tabula **“loma”** sastāv no 2 laukiem un ir savienota ar tabulu “lietotajs” ar saiti viens-pret-daudziem,
- tabula **“bernsvecaks”** sastāv no 3 laukiem un ir starptabula unārā tabulas “lietotajs” pašas ar sevi relācijā,
- tabula **“kostimi”** sastāv no 5 laukiem un ir savienota ar tabulu “lietotajs” ar saiti daudzi-pret-daudziem,
- tabula **“audzēkniskostimi”** sastāv no 4 laukiem un ir starptabula daudzi-pret-daudziem relācijā starp tabulām “lietotajs” un “kostimi”

- tabula **“grupa”** sastāv no 3 laukiem un ir savienota ar tabulu “lietotajs” ar saiti daudzi-pret-daudziem,
- tabula **“pedagogsgrupa”** sastāv no 4 laukiem un ir starptabula daudzi-pret-daudziem relācijā starp tabulām “lietotajs” un “grupa”,
- tabula **“audzeknisgrupa”** sastāv no 4 laukiem un ir starptabula daudzi-pret-daudziem relācijā starp tabulām “lietotajs” un “grupa”,
- tabula **“filiale”** sastāv no 6 laukiem un ir savienota ar tabulu “grupa” ar saiti viens-pret-daudziem,
- tabula **“numurs”** sastāv no 7 laukiem un ir savienota ar tabulu “kostimi” ar saiti viens-pret-daudziem kā arī ar tabulām “lietotajs” un “koncerts” ar saiti daudzi-pret-daudziem
- tabula **“pedagogsnumurs”** sastāv no 3 laukiem un ir starptabula daudzi-pret-daudziem relācijā starp tabulām “lietotajs” un “numurs”,
- tabula **“audzeknisnumurs”** sastāv no 3 laukiem un ir starptabula daudzi-pret-daudziem relācijā starp tabulām “lietotajs” un “numurs”,
- tabula **“koncerts”** sastāv no 6 laukiem un ir savienota ar tabulu “numurs” ar saiti daudzi-pret-daudziem
- tabula **“koncertsnumurs”** sastāv no 4 laukiem un ir starptabula daudzi-pret-daudziem relācijā starp tabulām “koncerts” un “numurs”,

Tabula “lietotajs” satur reģistrētu lietotāju datus: personas kodu, kas ir unikāls katram cilvēkam, tapēc tiks izmantots kā primāra atslēga, vārdu un uzvārdu, reģistrācijas e-pastu un parole, tālruni, dzimšanas dienu, kā arī lietotājam piesaistītas lomas id.

1.tabula

Tabulas **“lietotajs”** struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	personasKods	CHAR	12	Lietotāja personas kods
2.	Vards	VARCHAR	30	Lietotāja vārds
3.	Uzvards	VARCHAR	30	Lietotāja uzvārds
4.	Epasts	VARCHAR	30	Lietotāja reģistrācijas e-pasts
5.	Parole	VARCHAR	30	Lietotāja reģistrācijas parole
6.	Talrunis	CHAR	12	Lietotāja tālrunis
7.	DzimDiena	DATE	-	Lietotāja dzimšanas dienas datums
8.	LomaID	INT	-	Lietotājam piesaistītas lomas id

Tabula “loma” satur informāciju par sistēmas lietotāju lomām: lomas unikālo identifikātoru un tās nosaukumu.

2.tabula

Tabulas “**loma**” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	LomaID	INT	-	Lomas identifikātors
2.	LomasNosaukums	VARCHAR	30	Lomas nosaukums (“Administrators”, “Pedagogs”, “Audzēknis”, “Vecāks”)

Tabula “bernsvecaks” satur informāciju par lietotāju ar lomu “vecāks” bērniem un lietotāju ar lomu “audzēknis” vecākiem: ieraksta unikālo identifikātoru, bērna un vecāka personas kodus.

3.tabula

Tabulas “**bernsvecaks**” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	BernsVecaksID	INT	-	Ieraksta identifikātors
2.	personasKods	CHAR	12	Lietotāja ar lomu “Bērns” personas kods
3.	personasKods	CHAR	12	Lietotāja ar lomu “Vecāks” personas kods

tabula “kostimi” satur informāciju par sistēmā pieejamiem kostīmiem: identifikātoru, kostīma izmēru, nosaukumu, krāsu un tā attēlu.

4.tabula

Tabulas “**kostimi**” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	KostimiID	INT	-	Ieraksta identifikātors
2.	Nosaukums	VARCHAR	30	Kostīma nosaukums
3.	Krasa	VARCHAR	30	Kostīma krāsas nosaukums
4.	Izmers	VARCHAR	20	Kostīma izmērs
5.	Attels	BLOB	-	Kostīma attēls

Tabula “audzeknikostimi” satur informāciju par konkrētam lietotājam ar lomu “audzēknis” piesaistītiem kostīmiem: identifikatoru, kostīma piešķiršanas datumu, kostīma identifikatoru un audzēkņa personas kodu.

5.tabula

Tabulas “audzeknikostimi” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	AudzKostID	INT	-	Ieraksta identifikators
2.	PiesDatums	DATE	-	Datums, kad kostīms tika pišķirts lietotājam
3.	KostimiID	INT	-	Piešķirta kostīma identifikators
4.	AudzeknaKostimaPersonasKods	INT	-	Audzēkņa, kuram tika piešķirts kostīms personas kods

Tabula “grupa” satur informāciju par grupu: nosaukumu, kas skolas ietvaros ir unikāls, tapēc var kalpot kā unikālais identifikators, nodarbību grafiku un filiāles, kurā šī grupa atrodas ID.

6.tabula

Tabulas “grupa” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	GrupasNosaukums	VARCHAR	5	Grupas nosaukums
2.	Grafiks	TINYTEXT	-	Grupas nodarbību norises grafiks
3.	FilialeID	INT	-	Grupas atrašanas filiāles ID

Tabula “pedagogsgrupa” satur informāciju par lietotājiem ar lomu “pedagogs” piešķirtajām grupām: identifikatoru, nodarbību skaitu nedēļā, pedagoga personas kodu un grupas nosaukumu.

7.tabula

Tabulas “PedagogsGrupa” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	PedagogsGrupaID	INT	-	Ieraksta identifikators
2.	NodSk	INT	-	Nodarbību skaits, ko pedagogs novada konkrētajai grupai vienā nedēļā
3.	PedagogaPersonasKods	CHAR	12	Pedagoga personas kods
4.	GrupasPedagogaNosaukums	VARCHAR	5	Grupas nosaukums

Tabula “audzeknisgrupa” satur informāciju par grupu, kurai pieder lietotājs ar lomu “audzēknis”: identifikātoru, audzēkņa pieņemšanas datumu, audzēkņa personas kodu un grupas nosaukumu.

8.tabula

Tabulas “audzeknisgrupa” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	AudzeknisGrupaID	INT	-	Ieraksta identifikātors
2.	PienDatums	DATE	-	Datums, kurā audzēknis tika pieņemts grupā
3.	AudzeknaGrupasPersonasKods	CHAR	12	Audzēkņa personas kods
4.	GrupasAudzeknaNosaukums	VARCHAR	5	Grupas nosaukums

Tabula “filiale” satur informāciju par skolas filiāli: identifikātoru, atrašanas valsti, pilsētu, rajonu, ielu, ēku un pasta indeksu.

9.tabula

Tabulas “filiale” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	FilialeID	INT	-	Ieraksta identifikātors
2.	Valsts	VARCHAR	20	Filiāles atrašanas valsts
3.	Pilseta	VARCHAR	20	Filiāles atrašanas pilsēta
4.	Iela	VARCHAR	20	Filiāles atrašanas iela
5.	Eka	VARCHAR	10	Filiāles atrašanas ēka
6.	Indekss	CHAR	7	Filiāles atrašanas vietas pasta indekss

Tabula “numurs” satur informāciju par koncertos esošiem numuriem: numura nosaukumu, kārtas numuru, garumu, mūziku, horeogrāfijas stilu un izmantojama kostīma numuru un izpildītāju skaitu.

10.tabula

Tabulas “numurs” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	NumursID	INT	-	Numura identifikātors
2.	NumuraNosaukums	VARCHAR	30	Numura nosaukums
3.	Garums	VARCHAR	6	Numura garums
4.	Muzika	VARCHAR	30	Mūzikas kompozīcijas nosaukums
5.	Horeografija	VARCHAR	30	Horeografijas stila nosaukums
6.	IzpildītājuSkaitis	INT	-	Izpildītāju skaits
7.	KostimiIDnumurs	INT	-	Izmantotā kostīma ID

Tabula “pedagogsnumurs” satur informāciju par lietotājiem ar lomu “pedagogs” veidotiem numuriem: pedagoga personas kodu un numuru identifikātoru.

11.tabula

Tabulas “**pedagogsnumurs**” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	PedagogsNumursID	INT	-	Ieraksta identifikātors
2.	PedagogaNumuraPersonas Kods	CHAR	12	Pedagoga personas kods
3.	NumursIDPedagogs	INT	-	Numura identifikators

Tabula “AudzeknisNumurs” satur informāciju par lietotājiem ar lomu “Audzeknis” veidotiem numuriem: audzekņa personas kodu un numuru identifikātoru.

12.tabula

Tabulas “**audzeknisnumurs**” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	AudzeknisNumursID	INT	-	Ieraksta identifikātors
2.	AudzekņaPersonasKods	CHAR	12	Audzēkņa personas kods
3.	NumursIDAudzeknis	INT	-	Numura identifikators

Tabula “koncerts” satur informāciju par koncertiem: koncerta nosaukumu, datumu un norises vietu.

13.tabula

Tabulas “**koncerts**” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	KoncertsID	INT	-	Koncerta identifikātors
2.	Nosaukums	VARCHAR	30	Koncerta nosaukums
3.	Datums	DATE	-	Koncerta datums
4.	Vieta	VARCHAR	30	Koncerta norises vieta
5.	Saite	VARCHAR	60	Saite uz biļešu pārdošanas vietu
6.	Ilgums	VARCHAR	20	Koncerta garums

Tabula “koncertsnumurs” satur informāciju par koncertos iekļautiem numuriem: koncerta identifikācijas numuru un numuru identifikātoru.

14.tabula

Tabulas “**koncertsnumurs**” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	KoncertsNumursID	INT	-	Ieraksta identifikators
2.	KoncertsIDKoncNumurs	INT	-	Koncerta identifikators
3.	KartasNumurs	INT	-	Kārtas numurs koncerta programmā
4.	NumursIDKoncNumurs	INT	-	Numura identifikators

6. LIETOTĀJA CEĻVEDIS

6.1. Sistēmas prasības aparatūrai un programmatūrai

Baletskolas kostīmu uzskaites sistēma ir tīmekļa lietotne, kas ir piemērota ērtai lietošanai lielākajā daļā ierīču, ja tās atbilst sekojošām prasībām:

- Operētājsistēma:
 - Windows 10 vai jaunāka versija;
 - macOS 10.15 vai jaunāka versija;
 - jebkura moderna Linux versija.
- Interneta Pārlūks:
 - Google Chrome 89 vai jaunāka versija;
 - Mozilla Firefox 86 vai jaunāka versija;
 - Safari 14 vai jaunāka versija;
 - Microsoft Edge 89 vai jaunāka versija.

Ir jāņem vērā, ka lai ierīces spētu efektīvi darboties ar baletskolas kostīmu uzskaites sistēmu, nodrošinot lietotājiem optimālu lietotāja pieredzi ir jāparliecinās, ka izmantojamais pārlūks atbalsta JavaScript.

6.2. Sistēmas instalācija un palaišana

Lai instalētu un palaistu sistēmu lietotājam vispirms ir jāparupējās par sekojoša programmnodrošinājuma instalēšanu uz izmantojamās ierīces:

- Visual Studio Code
- Laragon (vai cits lokālās izstrādes serveris)
- Node.js
- Composer
- Git

1. Projekta klonēšana no GitHub:

1.1. Atveriet terminālu un izpildiet komandu (skat. 14.att. Git Clone);

```
git clone https://github.com/pikcrvt-prakse/dvb-2024-KatrinaGandzjuka.git
```

14.att. Git Clone

1.2. Ieejiet projekta mapē (skat. 15.att. Projekta mape).

```
cd dvb-2024-KatrinaGandzjuka/BaletskolasIntranets
```

15.att. Projekta mape

2. Nepieciešamu rīku instalēšana:

2.1. Instalējiet Composer (skat. 16.att. Composer);

```
composer install
```

16.att. Composer

2.2. Instalējiet npm (skat. 17.att. Npm).

```
npm install
```

17.att. Npm

3. Konfigurācijas faila iestatīšana:

3.1. Izveidojiet .env failu no .env.example (skat. 18.att. Env);

```
cp .env.example .env
```

18.att. Env

3.2. Atveriet .env failu un iestatiet datubāzes un citu konfigurāciju informāciju.

4. Datubāzes migrācijas:

4.1. Palaidiet datubāzes migrācijas, lai izveidotu tabulas (skat. 19.att. Migrācija).

```
php artisan migrate
```

19.att. Migrācija

5. Projekta palaišana lokālajā serverī:

5.1. Palaidiet Laravel lokālo serveri (skat. 20.att. Lokālais serveris);

```
php artisan serve
```

20.att. Lokālais serveris

5.2. Atveriet pārlūkprogrammu un ievadiet adresi (skat. 21.att. Adrese).

```
127.0.0.1:8000/login
```

21.att. Adrese

6. Lietotāju pieslēgšanas dati:

6.1. administrātoram var piekļūt, izmantojot personas kodu 180681-18171 un paroli Sila!123;

- 6.2. pedagoga profilam var piekļūt, izmantojot personas kodu 360201-27845 un paroli Kalnins!123;
- 6.3. audzēkņa profilam var piekļūt, izmantojot personas kodu 160906-27839 un paroli Silina!123;
- 6.4. vecāka profilam var piekļūt, izmantojot personas kodu 370301-27846 un paroli Ozola!123.

6.3. Programmas apraksts

Autentifikācija.

Atvērot tīmekļa lietotni pārlūkprogrammā, lietotājam tiek piedāvāts ievadīt savu personas kodu un paroli, ko viņam piešķir administrators (skat. 22.att.).

The screenshot shows a login interface with a blue header labeled 'Pieslēgšanās'. Below the header, there are two input fields: 'Personas kods:' with a placeholder 'Ievadiet personas kodu' and 'Parole:' with a placeholder 'Ievadiet paroli'. Below these fields is a link that says 'Neskaidrību gadījumā lūgums vērsties pie administrātora'. At the bottom left is a blue button labeled 'Ienākt'.

22.att. Pieslēgšanās forma

Pēc paroles un personas koda ievades lietotājam jānospiež poga "Ienākt", ja visi dati tika ievadīti korekti, lietotājs tiks pāradresēts uz viņa lomas funkcionalitātei piemērotu lapu.

Administrātora funkcionalitāte, jauna lietotāja pievienošana.

Veiksmīgās autentifikācijas gadījumā lietotājs ar administrātoru lomu tiek pāradresēts uz jauna lietotāja pievienošanas lapu (skat. 23.att.).

The screenshot shows a form for adding a new user. The header is 'Lietotāji' with sub-headers 'tēpi', 'grupas', 'numuri', 'koncērti'. Below the header are links 'pievienot lietotāju' and 'lietotāju saraksts'. The form contains several input fields: 'Personas kods:' (placeholder: 'Iem. 160906-27839'), 'Vārds:' (placeholder: 'Iem. Alise'), 'Uzvārds:' (placeholder: 'Iem. Silina'), 'E-pasts:' (placeholder: 'Iem. alisesilina@gmail.com'), 'Parole:' (placeholder: 'Iem. Silina!123'), 'Tālrunis:' (placeholder: 'Iem. +37198456763'), 'Dzimšanas diena:' (placeholder: 'mm/dd/yyyy'), and 'Loma:' (placeholder: 'Audzēknis'). At the bottom is a blue button labeled 'Pievienot lietotāju'.

23.att. Jauna lietotāja pievienošanas forma

Pēc jauna lietotāja informācijas ievades administratoram ir nepieciešams uzpiezt pogu "Pievienot lietotāju", lai lietotājs tiktu pievienots sarakstam.

Lietotāju saraksts.

Uzklikšķinot uz sadaļas "lietotāju saraksts" izvelnē, kas atrodas lapas augšējā pusē, administratoram ir iespēja atvert sarakstu ar visiem lietotājiem (skat. 24.att.)

Lietotāji							
tārpi grupas numuri koncerti							
pievienot lietotāju lietotāju saraksts							
audzēkņi vecāki pedagogi							
Loma	Vārds un uzvārds	Personas Kods	Tālrunis	E-pasts	Dzimšanas diena	Rediģēšana	Dzēšana
audzēkņis	Eva Kalnina	150906-27861	+37112345601	evakalnina@gmail.com	2006-09-15	Rediģēt	Dzēst
audzēkņis	Alise Silina	160906-27839	+37198456763	alisesilina@gmail.com	2006-09-16	Rediģēt	Dzēst
audzēkņis	Artis Bērziņš	160907-27862	+37112345602	artisberzins@gmail.com	2007-09-16	Rediģēt	Dzēst
audzēkņis	Elza Ozoliņa	170908-27863	+37112345603	elzaozolina@gmail.com	2008-09-17	Rediģēt	Dzēst
audzēkņis	Jānis Bērziņš	171006-27840	+37198456764	janisberzins@gmail.com	2006-10-17	Rediģēt	Dzēst
audzēkņis	Līga Kalnina	180106-27841	+37198456765	ligakalnina@gmail.com	2006-01-18	Rediģēt	Dzēst
audzēkņis	Rūdolfs Liepiņš	180909-27864	+37112345604	rudolfsliepins@gmail.com	2009-09-18	Rediģēt	Dzēst
audzēkņis	Pēteris Ozoliņš	190406-27842	+37198456766	peterisozolins@gmail.com	2006-04-19	Rediģēt	Dzēst
audzēkņis	Guna Klavina	190910-27865	+37112345605	gunaklavina@gmail.com	2010-09-19	Rediģēt	Dzēst
audzēkņis	Maija Liepa	200206-27843	+37198456767	maijalepa@gmail.com	2006-02-20	Rediģēt	Dzēst
audzēkņis	Raitis Zariņš	200911-27866	+37112345606	raitiszarins@gmail.com	2011-09-20	Rediģēt	Dzēst
audzēkņis	Anna Liepiņa	210706-27844	+37198456778	annaliepina@gmail.com	2007-07-21	Rediģēt	Dzēst

24.att. Lietotāju saraksts

Lai filtrētu lietotājus pēc lomas, ir jāizmanto izvēlne, kas atrodas lapas augšējā pusē, uzklikšķinot uz vēlamās lomas (skat. 25.att.).

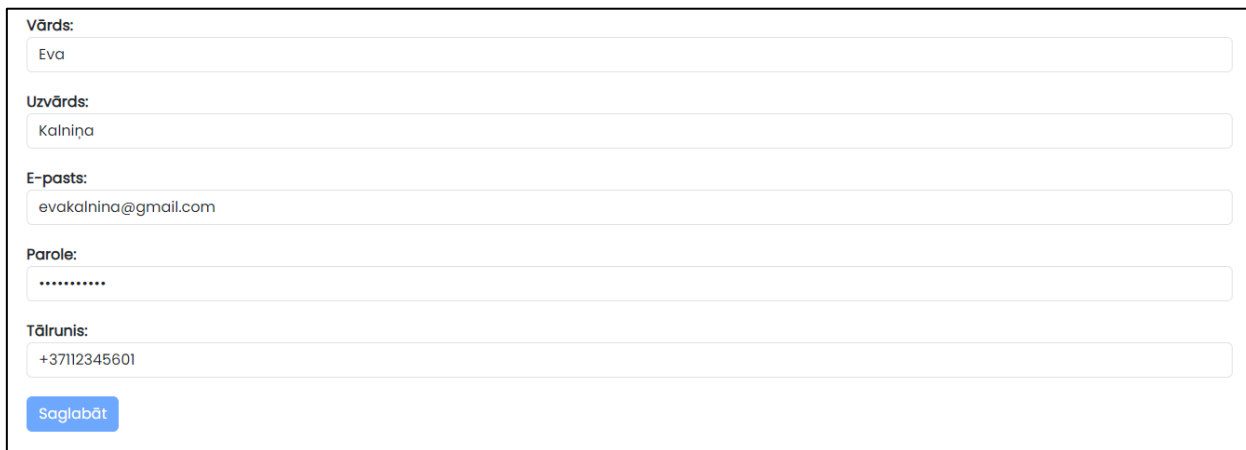


25.att. Lomas izvēlne

Pēc vēlamās lomas izvēles saraksts tiks filtrēts, parādot tikai to lietotāju datus, kas atbilst konkrētai lomai.

Lietotāja rediģēšana.

Lietotāja datu rediģēšanai ir jāizmanto poga "rediģēt", kas atrodas lietotāju tabulas labajā pusē. Nospiežot šo pogu, administrātors var piekļūt lietotāja datu rediģēšanas formai (skat. 26.att.).



Vārds:
Eva

Uzvārds:
Kalniņa

E-pasts:
evakalniņa@gmail.com

Parole:
.....

Tālrunis:
+37112345601

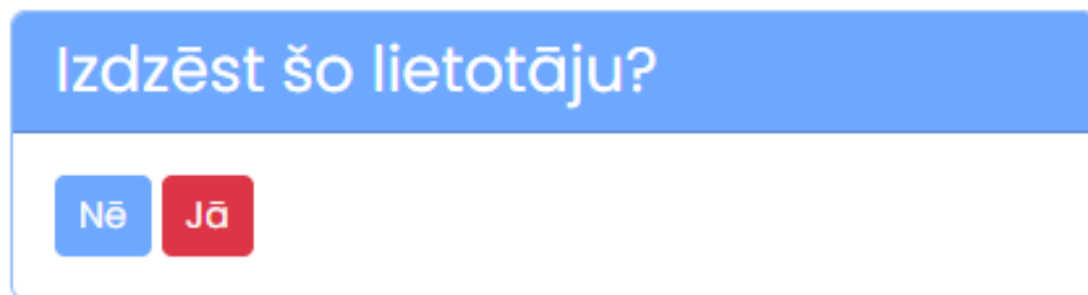
Saglabāt

26.att. Lietotāju datu rediģēšanas forma

Lai saglabātu veiktās izmaiņas, lietotājam jāpiespiež poga "Saglabāt", lai rediģēta lietotāja dati tiktu atjaunoti.

Lietotāja dzēšana.

Ja administrātors vēlās nodzēst lietotāju no saraksta, viņam jāpiespiež poga "Dzēst" lietotāju saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pārdresēts lapā, kur dzēšana būs jāapstiprināvai jānoliedz piespiežot pogu "Jā" vai "Nē" (skat. 27.att.).



Izdzēst šo lietotāju?

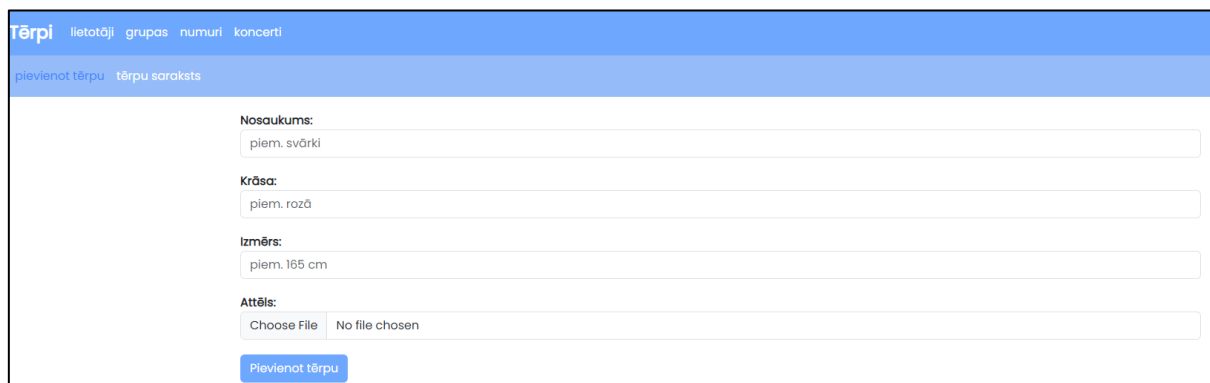
Nē Jā

27.att. Lietotāja dzēšanas apstiprināšana

Pēc dzēšanas apstiprināšanas lietotājs tiks izdzēsts no saraksta, ja dzēšana tiks noliegta, tas netiks izdarīts. Abos gadījumos administrātors tiks atgriezts uz lietotāja saraksta pārskata lapu.

Jauna tērpa pievienošana.

Izvelnē, kas atrodas lapas augšējā daļā, izvēloties sadaļu "tērpi", administrātors tiks pārdresēts uz jauna tērpa pievienošanas formu (skat. 28.att.).

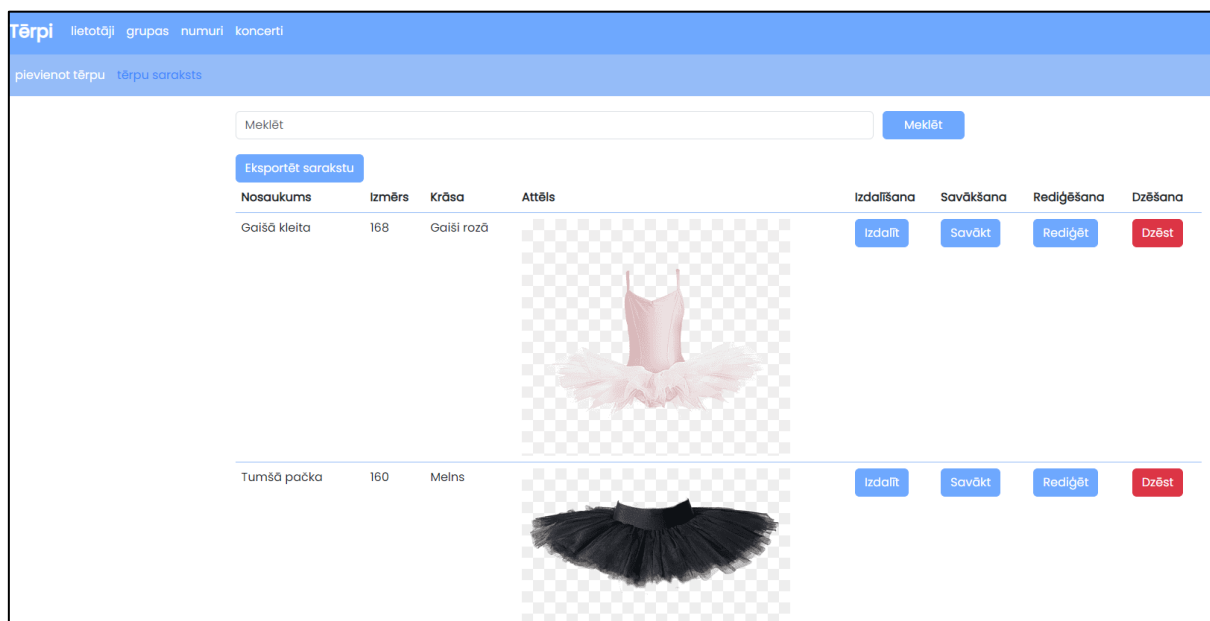




28.att. Jauna tērpa pievienošanas forma

Pēc jauna tērpa informācijas ievades administrātoram ir nepieciešams uzpiezt pogu "Pievienot tērpu", lai tērps tiktu pievienots sarakstam.

Tērpu saraksts.

Uzklikšķinot uz sadaļas "tērpu saraksts" izvelnē, kas atrodas lapas augšējā pusē, administrātoram ir iespēja atvert sarakstu ar visiem tērpiem (skat. 29.att.).



Nosaukums	Izmērs	Krāsa	Attēls	Izdalīšana	Savākšana	Rediģēšana	Dzēšana
Gaišā kleita	168	Gaiši rozā		Izdalīt	Savākt	Rediģēt	Dzēst
Tumsā pakča	160	Melns		Izdalīt	Savākt	Rediģēt	Dzēst

29.att. Tērpu saraksts

Lai veiktu tērpa meklēšanu, jāizmanto meklēšanas forma, kas atrodas lapas augšējā daļā. Formā var ierakstīt jebkuru atslēgvārdu, pēc kura tiks nodrošināta tērpa meklēšana (skat. 30.att.).



30.att. Meklēšanas forma

Pēc meklējamā atslēgvārda norādīšanas lietotājam jāpiespiež poga "Meklēt", lai tiktu atlasīti tie tērpi, kuru dati satur meklējamo atslēgvārdu.

Tērpu saraksta eksportēšana.

Lai eksportētu visu noliktavā esošo tērpu sarakstu kā PDF failu, ir jāpiespiež poga "Eksportēt sarakstu", kas atrodas zem meklēšanas formas (skat. 31.att.).

Eksportēt sarakstu

31.att. Tērpu saraksta eksporta poga

Pēc pogas uzspiešanas PDF fails tiks automātiski lejupielādēts uz ierīces, to varēs atrast sadaļā "Downloads" (skat. 32.att.).

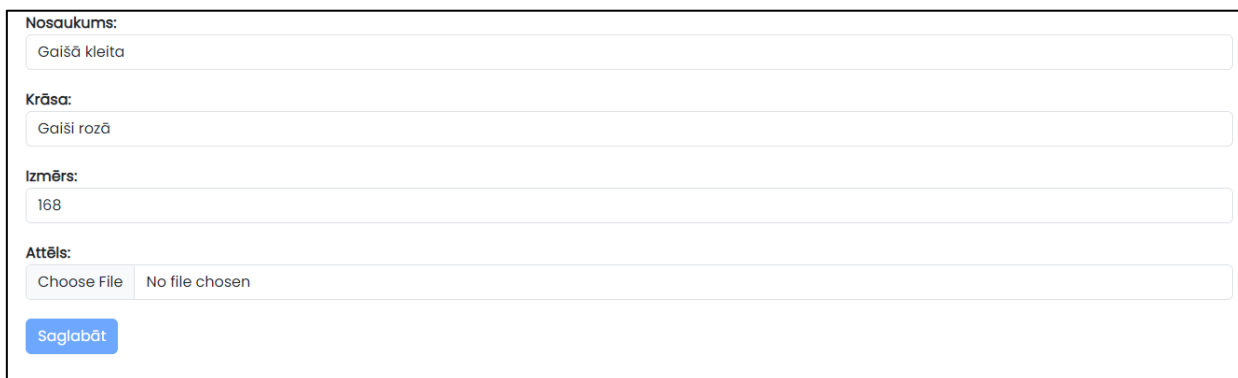
 Grande Ballet Studio			
Tērpu saraksts			
Pieejamo tērpu skaits: 8			
Nosaukums	Krāsa	Izmērs	Attēls
Gaišā kleita	Gaiši rozā	168	
Tumšā pačķa	Melns	160	
Bērnu tilla svārki	Spilgti rozā	xs	
Bērnu svārki	Tumši zils	120	
Garas bikses	Melns	M	
Spilgts kreklis	Sarkans	S	

32.att. Tērpu saraksta PDF Fails

Turpmāk šo failu varēs izprintēt vai kopīgot jebkādā lietotājam ērtajā veidā.

Tērpa rediģēšana.

Tērpu datu rediģēšanai ir jāizmanto poga "rediģēt", kas atrodas tērpu tabulas labajā pusē. Nospiežot šo pogu, administrātors var piekļūt tērpu datu rediģēšanas formai (skat. 33.att.).

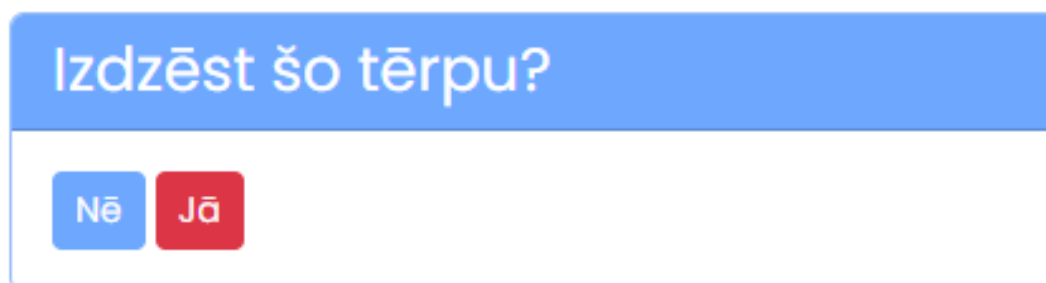


33.att. Tērpu datu rediģēšanas forma

Lai saglabātu veiktās izmaiņas, lietotājam jāpiespiež poga "Saglabāt", lai rediģēta tērpa dati tiktu atjaunoti.

Tērpa dzēšana.

Ja administrātors vēlās nodzēst tērpu no saraksta, viņam jāpiespiež poga "Dzēst" tērpu saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā, kur dzēšana būs jāapstiprināvai jānoliedz piespiežot pogu "Jā" vai "Nē" (skat. 34.att.).



34.att. Tērpa dzēšanas apstiprināšana

Pēc dzēšanas apstiprināšanas tērps tiks izdzēsts no saraksta, ja dzēšana tiks noliegta, tas netiks izdarīts. Abos gadījumos administrātors tiks atgriezts uz tērpu saraksta pārskata lapu.

Tērpa izdalīšana.

Ja administrātors vēlās izdalīt tērpu audzēknim, viņam jāpiespiež poga "Izdalīt" tērpu saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar lietotāju sarakstu, kur lai izsniegt tērpu ir jāpiespiež poga "Izsniegt" (skat. 35.att.).

Vārds un uzvārds	Grupa	Izsniegšana
Artis Bērziņš	P4	Izsniegt
Elza Ozoliņa	P4	Izsniegt
Jānis Bērziņš	P1	Izsniegt
Rūdolfs Liepiņš	P4	Izsniegt
Pēteris Ozoliņš	P1	Izsniegt
Guna Kļaviņa	P4	Izsniegt
Maija Līepa	P1	Izsniegt
Raitis Zariņš	P4	Izsniegt
Anna Liepiņa	P2	Izsniegt
Ilze Vētra	P2	Izsniegt
Kaspars Kļaviņš	P5	Izsniegt

35.att. Audzēkņu saraksts tērpa izdalīšanai

Pēc izsniegšanas pogas piespiešanas, tērps tiks izsniegts audzēknim un tas pazudīs no pieejamo audzēkņu saraksta.

Tērpa savākšana.

Ja administrātors vēlās savākt tērpu no audzēkņa, viņam jāpiespiež poga "Savākt" tērpu saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar lietotāju sarakstu, kur lai savākt tērpu ir jāpiespiež poga "Savākt" (skat. 36.att.).

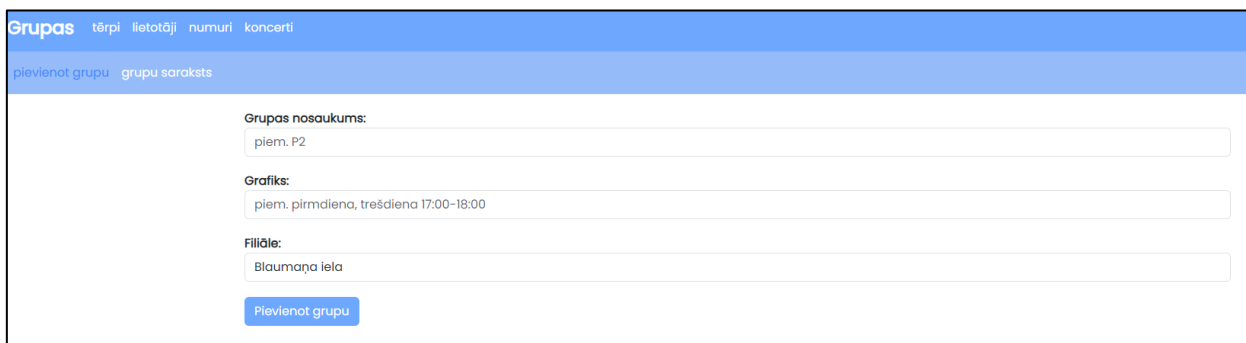
Vārds un uzvārds	Grupa	Savākšana
Eva Kalniņa	P4	Savākt
Līga Kalniņa	P1	Savākt
Alise Siliņa	P1	Savākt
Inese Ābele	P4	Savākt
Guna Kļaviņa	P4	Savākt

36.att. Audzēkņu saraksts tērpa savākšanai

Pēc savākšanas pogas piespiešanas, tērps tiks savākts audzēknim un tas pazudīs no audzēkņu saraksta.

Jaunas grupas pievienošana.

Izvelnē, kas atrodas lapas augšējā daļā, izvēloties sadaļu "grupas", administrātors tiks pārdresēts uz jaunas grupas pievienošanas formu (skat. 37.att.).



37.att. Jaunas grupas pievienošanas forma

Pēc jaunas grupas informācijas ievades administrātoram ir nepieciešams uzpiezt pogu "Pievienot grupu", lai grupa tiktu pievienota sarakstam.

Grupu saraksts.

Uzklikšķinot uz sadaļas "grupu saraksts" izvelnē, kas atrodas lapas augšējā pusē, administrātoram ir iespēja atvert sarakstu ar visām grupām (skat. 38.att.).



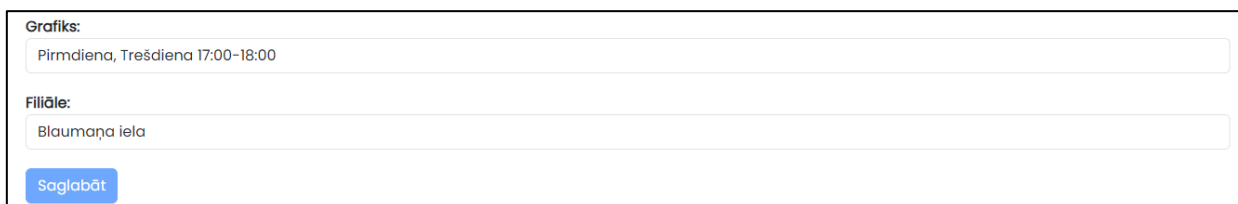
Nosaukums	Grafiks	Filiale	Audzēkņu skaits	Dalībnieki	Rediģēšana	Dzēšana
P1	Pirmdiena, Trešdiena 17:00-18:00	Blaumaņa	5	Saraksts	Rediģēt	Dzēst
P2	Otrdiena, Trešdiena, Ceturtdiena 17:00-18:00	Blaumaņa	6	Saraksts	Rediģēt	Dzēst
P3	Trešdiena, Piekdiena 17:00-18:00	Blaumaņa	4	Saraksts	Rediģēt	Dzēst
P4	Pirmdiena, Trešdiena 18:00-19:00	Saharova	7	Saraksts	Rediģēt	Dzēst
P5	Otrdiena, Ceturtdiena 18:00-19:00	Blaumaņa	5	Saraksts	Rediģēt	Dzēst

38.att. Grupu saraksts

Tagad administrātoram ir pieejams saraksts ar visu studijā esošo grupu sarakstu.

Grupas rediģēšana.

Grupu datu rediģēšanai ir jāizmanto poga "rediģēt", kas atrodas grupu tabulas labajā pusē. Nospiežot šo pogu, administrātors var piekļūt grupas datu rediģēšanas formai (skat. 39.att.).

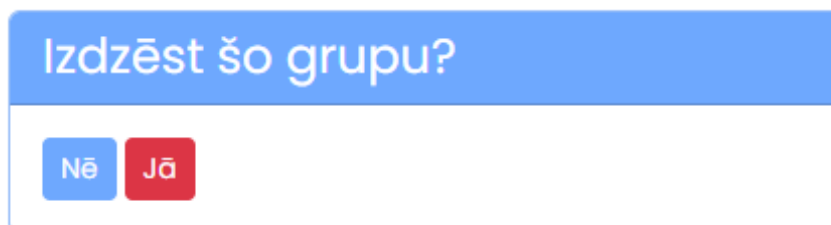


39.att. Grupas datu rediģēšanas forma

Lai saglabātu veiktās izmaiņas, lietotājam jāpiespiež poga "Saglabāt", lai rediģēta grupas dati tiktu atjaunoti.

Grupu dzēšana.

Ja administrātors vēlās nodzēst grupu no saraksta, viņam jāpiespiež poga "Dzēst" grupu saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā, kur dzēšana būs jāapstiprināvai jānoliedz piespiežot pogu "Jā" vai "Nē" (skat. 40.att.).



40.att. Grupas dzēšanas apstiprināšana

Pēc dzēšanas apstiprināšanas grupa tiks izdzēsta no saraksta, ja dzēšana tiks noliegta, tas netiks izdarīts. Abos gadījumos administrātors tiks atgriezts uz grupu saraksta pārskata lapu.

Dalībnieku saraksts.

Ja administrātors vēlās piekļūt grupas audzēkņu sarakstam, viņam jāpiespiež poga "Saraksts" grupu saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar lietotāju sarakstu (skat. 41.att.).

Pievienot audzēkni Pievienot pedagogu				
Eksportēt sarakstu				
Loma	Vārds un uzvārds	Tālruna numurs	E-pasts	Dzēšana
audzēknis	Alise Silina	+37198456763	alisesilina@gmail.com	Dzēst no grupas
audzēknis	Jānis Bērziņš	+37198456764	janisberzins@gmail.com	Dzēst no grupas
audzēknis	Līga Kalnina	+37198456765	ligakalnina@gmail.com	Dzēst no grupas
audzēknis	Pēteris Ozoliņš	+37198456766	peterisozolins@gmail.com	Dzēst no grupas
audzēknis	Maija Liepa	+37198456767	maiialepa@gmail.com	Dzēst no grupas
pedagogs	Edgars Grīns	+37198456775	edgarsgrins@gmail.com	Dzēst no grupas

41.att. Audzēkņu saraksts grupā

Lai dzēst dalībniekus no grupas, administrātoram jāpiespiež poga "Dzēst no grupas", kas atrodas saraksta labajā pusē. Pēc pogas nospiešanas, dalībnieks tiks izdzēsts no grupas.

Grupās dalībnieku saraksta eksportēšana.


Lai eksportētu grupas dalībnieku sarakstu kā PDF failu, ir jāpiespiež poga "Eksportēt sarakstu", kas atrodas virs grupas dalībnieku saraksta (skat. 42.att.).

Eksportēt sarakstu

42.att. Grupas dalībnieku saraksta eksporta poga

Pēc pogas uzspiešanas PDF fails tiks automātiski lejupielādēts uz ierīces, to varēs atrast sadaļā "Downloads".

Kad lietotājs atvērs failu sev ērtajā veidā, viņš varēs iepazīties ar grupas dalībnieku un viņu vecāku sarakstu (skat. 43.att.).



P1

Grafiks: Pirmdiena, Trešdiena 17:00-18:00

Pasniedzējs: Edgars Grīns

Audzēkņu skaits: 5

Vārds	Uzvārds	Tālruna numurs	Vecāks (Vārds, Uzvārds, Tālrunis)
Alise	Siliņa	+37198456763	Inga Bērziņa (+37198456768)
Jānis	Bērziņš	+37198456764	Andris Kalniņš (+37198456769)
Līga	Kalniņa	+37198456765	Sandra Ozola (+37198456770)
Pēteris	Ozoliņš	+37198456766	Guna Kļava (+37198456771)
Maija	Liepa	+37198456767	Elīna Vītola (+37198456772)

43.att. Grupas dalībnieku saraksta PDF Fails

Turpmāk šo failu varēs izprintēt vai kopīgot gadījumos, kad ir nepieciešams zināt grupas audzēkņu un viņu vecāku kontaktinformāciju.

Audzēkņa pievienošana grupai.

Ja administrātors vēlās pievienot audzēkni grupai, viņam izvēlnē, kas atrodas lapas augšdaļā jāizvēlas sadaļa "pievienot audzēkni". Pēc pogas piespiešanas administrātors tiks pārdresēts lapā ar lietotāju sarakstu, kur lai pievienot audzēkni grupai ir jāpiespiež poga "Pievienot grupai" (skat. 44.att.).

Vārds un uzvārds	Vecums	Pievienošana
Eva Kalniņa	17	Pievienot grupai
Artis Bērziņš	16	Pievienot grupai
Elza Ozoliņa	15	Pievienot grupai
Rūdolfs Liepiņš	14	Pievienot grupai
Guna Kļaviņa	13	Pievienot grupai
Raitis Zariņš	12	Pievienot grupai
Anna Liepina	16	Pievienot grupai
Inese Ābele	11	Pievienot grupai
Ilze Vētra	15	Pievienot grupai
Kaspars Kļaviņš	10	Pievienot grupai

44.att. Audzēkņu saraksts pievienošanai grupā

Pēc pievienošanas pogas piespiešanas, audzēknis tiks pievienots grupai un tas pazudīs no pieejamo audzēkņu saraksta.

Pedagoga pievienošana grupai.

Ja administrātors vēlās pievienot grupai, pedagogu viņam izvēlnē, kas atrodas lapas augšdaļā jāizvēlas sadaļa "pievienot pedagogu". Pēc pogas piespiešanas administrātors tiks pārdresēts lapā ar pedagogu sarakstu, kur lai pievienot pedagogu grupai ir jāievada nodarbību skaits, ko vadīs pedagogs un jāpiespiež poga "Pievienot grupai" (skat. 45.att.).

Nodarbību skaits nedēļā:	
<input type="text"/>	
Vārds un uzvārds	Pievienošana
Māris Jansons	Pievienot grupai
Ilze Krauze	Pievienot grupai

45.att. Pedagogu saraksts pievienošanai grupā

Pēc pievienošanas pogas piespiešanas, pedagogs tiks pievienots grupai un tas pazudīs no pieejamo pedagogu saraksta.

Jauna numura pievienošana.

Izvelnē, kas atrodas lapas augšējā daļā, izvēloties sadaļu "numuri", administrātors tiks pāradresēts uz jauna numura pievienošanas formu (skat. 46.att.).

Numuri

tērpi

lietotāji

grupas

koncerti

Pievienot numuru

Numuru saraksts

Numura nosaukums:

piem. Ziedu valsis

Garums:

piem. 2:30

Mūzika:

piem. Ziedu simfonija

Horeogrāfija:

piem. klasiskā deja

Izpildītāju skaits:

ievadiet izpildītāju skaitu

Kostīms:

Gaišā kleita

Pievienot numuru

46.att. Jauna numura pievienošanas forma

Pēc jauna numura informācijas ievades administrātoram ir nepieciešams uzpiezt pogu "Pievienot numuru", lai numurs tiktu pievienots sarakstam.

Numuru saraksts.

Uzklikšķinot uz sadaļas "numuru saraksts" izvelnē, kas atrodas lapas augšējā pusē, administrātoram ir iespēja atvert sarakstu ar visiem numuriem (skat. 47.att.).

Numuri

tērpi

lietotāji

grupas

koncerti

Pievienot numuru

Numuru saraksts

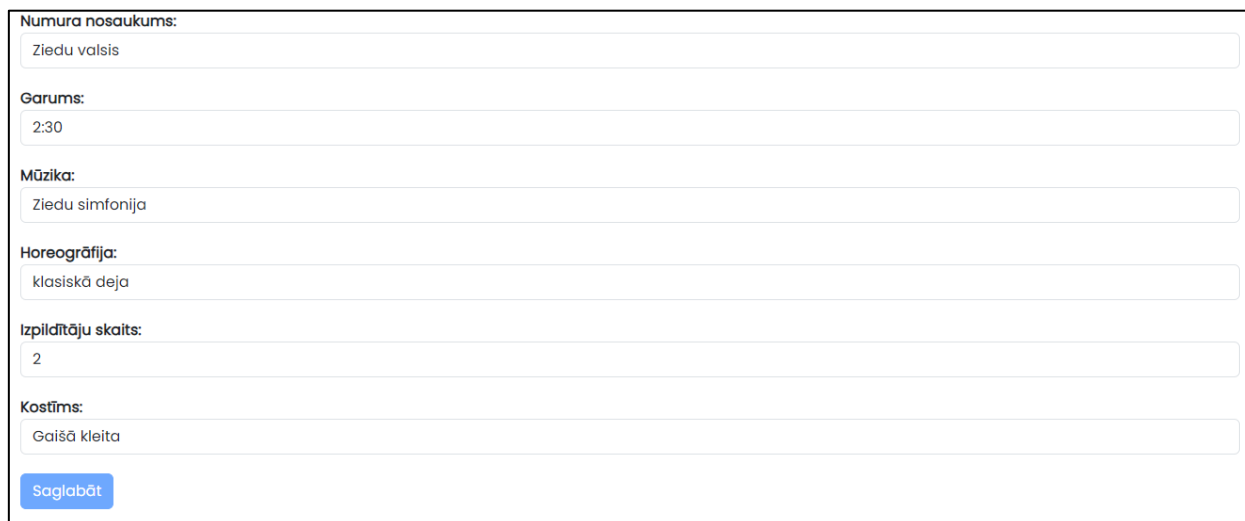
Nosaukums	Garums	Mūzika	Horeogrāfija	Izpildītāju skaits	Kostīms	Audzēkņi	Pedagogi	Rediģēšana	Dzēšana
Ziedu valsis	2:30	Ziedu simfonija	klasiskā deja	2	Gaišā kleita	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Saules deja	3:00	Saules rīts	moderna deja	2	Tumšā pačka	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Mēness serenāde	2:45	Mēness gaismā	ballets	6	Gaišā kleita	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Rudens lapa	3:15	Rudens sonāte	jazz	2	Bērnu svārki	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Jūras vējš	2:50	Jūras skaņas	contemporary	4	Gaišā kleita	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Kalnu ezeri	3:05	Kalnu ezeri	folk	3	Garas bikses	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Ziemas brīnums	2:40	Ziemas pasaka	klasiskā deja	3	Cepure	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Pavasara ziedi	3:10	Pavasara ziedi	moderna deja	4	Bērnu svārki	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Dzīvības upe	2:35	Dzīvības simfonija	jazz	4	Spilgts kreklis	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst
Sapņu pasaule	3:00	Sapņu dziesma	contemporary	5	Tumšā pačka	Izpildītāji	Horeogrāfi	Rediģēt	Dzēst

47.att. Numuru saraksts

Tagad administrātoram ir pieejams saraksts ar visiem studijai esošiem numuriem.

Numura rediģēšana.

Numura datu rediģēšanai ir jāizmanto poga "rediģēt", kas atrodas numuru tabulas labajā pusē. Nospiežot šo pogu, administrātors var piekļūt numura datu rediģēšanas formai (skat. 48.att.).

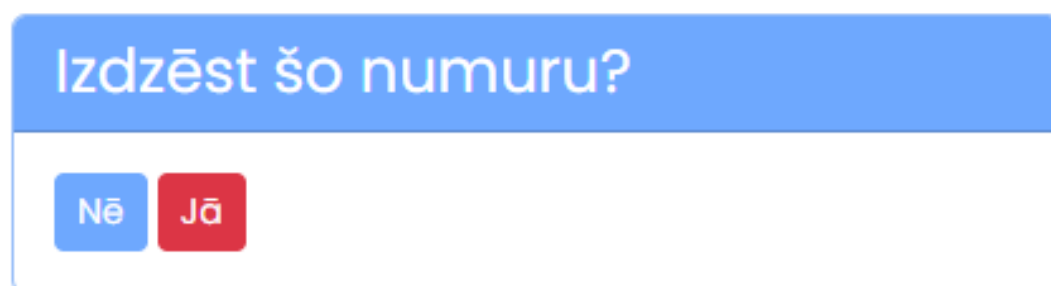


48.att. Numura datu rediģēšanas forma

Lai saglabātu veiktās izmaiņas, lietotājam jāpiespiež poga "Saglabāt", lai rediģēta numura dati tiktu atjaunoti.

Numuru dzēšana.

Ja administrātors vēlās nodzēst numuru no saraksta, viņam jāpiespiež poga "Dzēst" numuru saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā, kur dzēšana būs jāapstiprināvai jānoliedz piespiežot pogu "Jā" vai "Nē" (skat. 49.att.).



49.att. Numura dzēšanas apstiprināšana

Pēc dzēšanas apstiprināšanas grupa tiks izdzēsta no saraksta, ja dzēšana tiks noliegta, tas netiks izdarīts. Abos gadījumos administrātors tiks atgriezts uz numuru saraksta pārskata lapu.

Izpildītāju saraksts.

Ja administrātors vēlās piekļūt numura izpildītāju sarakstam, viņam jāpiespiež poga "Izpildītāji" numura saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar lietotāju sarakstu (skat. 50.att.).

Ziedu valsis <small>Pievienot audzēkni</small>			
Loma	Vārds un uzvārds	Grupa	Dzēšana
audzēknis	Eva Kalniņa	P4	<button>Dzēst no numura</button>
audzēknis	Alise Silina	P1	<button>Dzēst no numura</button>

50.att. Numura izpildītāju saraksts

Lai dzēst dalībniekus no numura, administrātoram jāpiespiež poga "Dzēst no numura", kas atrodas saraksta labajā pusē. Pēc pogas nospiešanas, dalībnieks tiks izdzēsts no numura.

Horeogrāfu saraksts.

Ja administrātors vēlās piekļūt numura horeogrāfu sarakstam, viņam jāpiespiež poga "Horeogrāfi" numura saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar lietotāju sarakstu (skat. 51.att.).

Ziedu valsis <small>Pievienot pedagogu</small>		
Loma	Vārds un uzvārds	Dzēšana
pedagogs	Ilze Krauze	<button>Dzēst no numura</button>

51.att. Numura horeogrāfu saraksts

Lai dzēst horeogrāfus no numura, administrātoram jāpiespiež poga "Dzēst no numura", kas atrodas saraksta labajā pusē. Pēc pogas nospiešanas, horeogrāfs tiks izdzēsts no numura.

Audzēkņa pievienošana numuram.

Ja administrātors vēlās pievienot audzēkni numuram, viņam izvelnē, kas atrodas lapas augšdaļā jāizvelās sadaļa "pievienot audzēkni". Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar lietotāju sarakstu, kur lai pievienot audzēkni numuram ir jāpiespiež poga "Pievienot numuram" (skat. 52.att.).

Vārds un uzvārds	Grupa	Pievienošana
Artis Bērziņš	P4	<button>Pievienot numuram</button>
Elza Ozoliņa	P4	<button>Pievienot numuram</button>
Jānis Bērziņš	P1	<button>Pievienot numuram</button>
Līga Kalniņa	P1	<button>Pievienot numuram</button>
Rūdolfs Liepiņš	P4	<button>Pievienot numuram</button>
Pēteris Ozoliņš	P1	<button>Pievienot numuram</button>
Guna Kļaviņa	P4	<button>Pievienot numuram</button>
Maija Liepa	P1	<button>Pievienot numuram</button>
Raitis Zariņš	P4	<button>Pievienot numuram</button>
Anna Liepina	P2	<button>Pievienot numuram</button>
Inese Ābele	P4	<button>Pievienot numuram</button>
Ilze Vētra	P2	<button>Pievienot numuram</button>

52.att. Audzēkņu saraksts pievienošanai numuram

Pēc pievienošanas pogas piespiešanas, audzēknis tiks pievienots numuram un tas pazudīs no pieejamo audzēkņu saraksta.

Pedagoga pievienošana numuram.

Ja administrātors vēlās pievienot numuram, pedagogu viņam izvelnē, kas atrodas lapas augšdaļā jāizvelās sadaļa "pievienot pedagogu". Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar pedagogu sarakstu, kur lai pievienot pedagogu numuram ir jāpiespiež poga "Pievienot numuram" (skat. 53.att.).

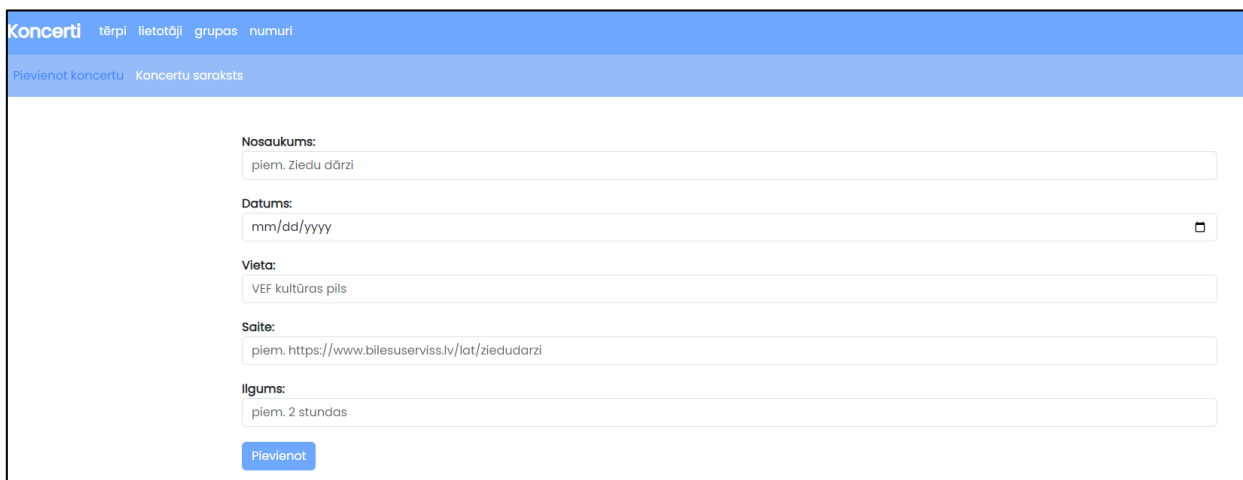
Vārds un uzvārds	Pievienošana
Māris Jansons	<button>Pievienot numuram</button>
Edgars Grīns	<button>Pievienot numuram</button>

53.att. Pedagogu saraksts pievienošanai numuram

Pēc pievienošanas pogas piespiešanas, pedagogs tiks pievienots numuram un tas pazudīs no pieejamo pedagogu saraksta.

Jauna koncerta pievienošana.

Izvelnē, kas atrodas lapas augšējā daļā, izvēloties sadaļu "koncerti", administrātors tiks pāradresēts uz jauna koncerta pievienošanas formu (skat. 54.att.).

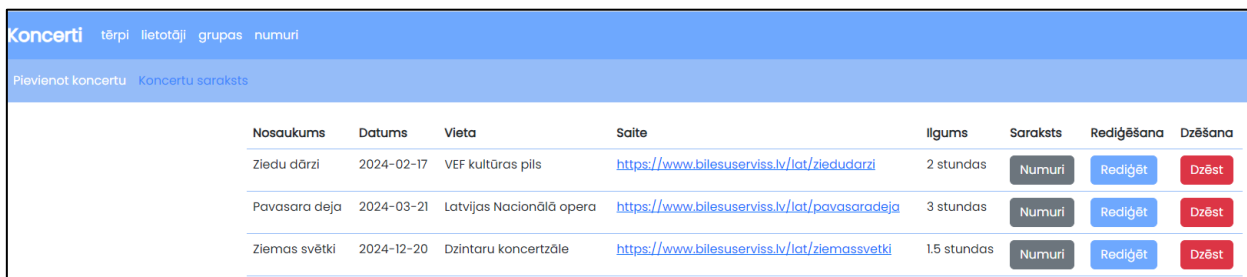


54.att. Jauna koncerta pievienošanas forma

Pēc jauna koncerta informācijas ievades administrātoram ir nepieciešams uzpiezt pogu "Pievienot koncertu", lai koncerts tiktu pievienots sarakstam.

Koncertu saraksts.

Uzklīkšķinot uz sadaļas "koncertu saraksts" izvelnē, kas atrodas lapas augšējā pusē, administrātoram ir iespēja atvert sarakstu ar visiem koncertiem (skat. 55.att.).



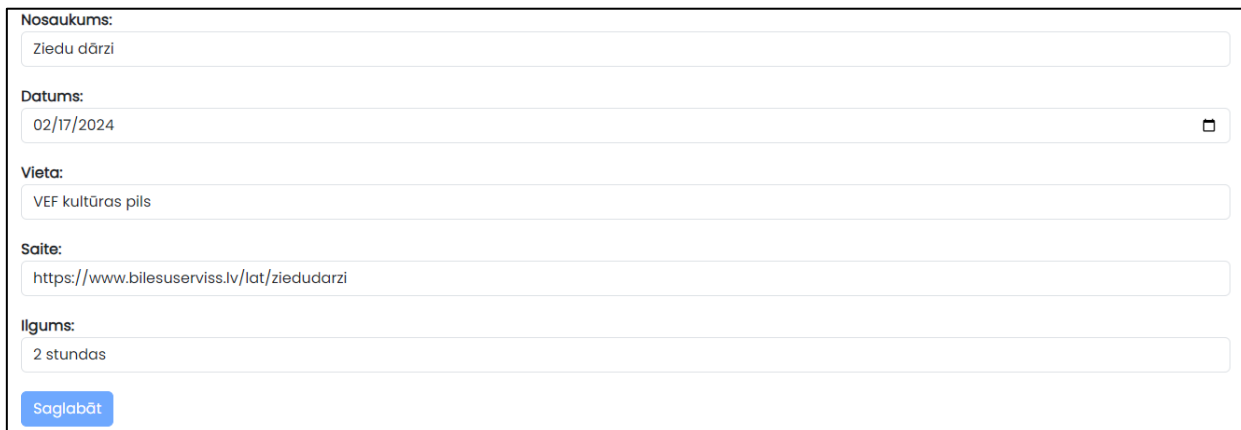
Nosaukums	Datums	Vieta	Saite	Ilgums	Saraksts	Rediģēšana	Dzēšana
Ziedu dārzi	2024-02-17	VEF kultūras pils	https://www.bilesuserviss.lv/lat/ziedudarzi	2 stundas	Numuri	Rediģēt	Dzēst
Pavasara deja	2024-03-21	Latvijas Nacionālā opera	https://www.bilesuserviss.lv/lat/pavasaradeja	3 stundas	Numuri	Rediģēt	Dzēst
Ziemas svētki	2024-12-20	Dzintaru koncertzāle	https://www.bilesuserviss.lv/lat/ziemassvetki	1.5 stundas	Numuri	Rediģēt	Dzēst

55.att. Koncertu saraksts

Tagad administrātoram ir pieejams saraksts ar visiem studijai esošiem koncertiem.

Koncerta rediģēšana.

Koncerta datu rediģēšanai ir jāizmanto poga "rediģēt", kas atrodas koncertu tabulas labajā pusē. Nospiežot šo pogu, administrātors var piekļūt koncerta datu rediģēšanas formai (skat. 56.att.).



Nosaukums:
Ziedu dārzi

Datums:
02/17/2024

Vieta:
VEF kultūras pils

Saite:
https://www.bilesuserviss.lv/lat/ziedudarzi

Ilgums:
2 stundas

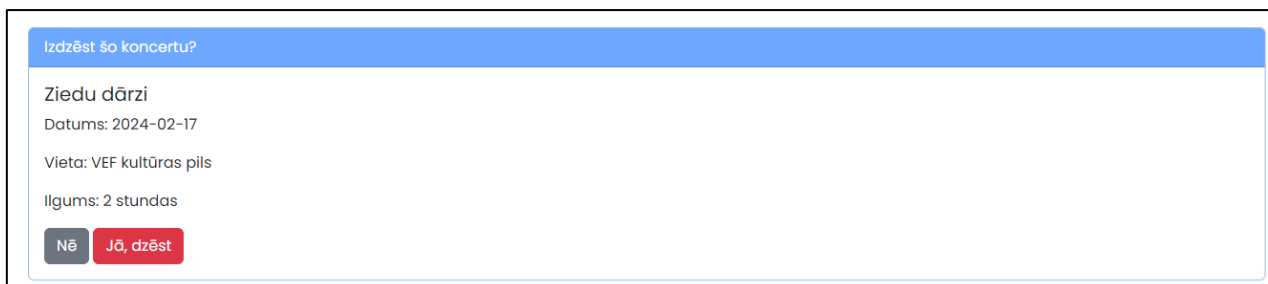
Saglabāt

56.att. Koncerta datu rediģēšanas forma

Lai saglabātu veiktās izmaiņas, lietotājam jāpiespiež poga "Saglabāt", lai rediģēta koncerta dati tiktu atjaunoti.

Koncerta dzēšana.

Ja administrātors vēlās nodzēst koncertu no saraksta, viņam jāpiespiež poga "Dzēst" koncertu saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā, kur dzēšana būs jāapstiprināvai jānoliedz piespiežot pogu "Jā" vai "Nē" (skat. 57.att.).



Izdzēst šo koncertu?

Ziedu dārzi
Datums: 2024-02-17
Vieta: VEF kultūras pils
Ilgums: 2 stundas

Nē Jā, dzēst

57.att. Koncerta dzēšanas apstiprināšana

Pēc dzēšanas apstiprināšanas koncerts tiks izdzēsta no saraksta, ja dzēšana tiks noliegta, tas netiks izdarīts. Abos gadījumos administrātors tiks atgriezts uz koncerta saraksta pārskata lapu.

Numuru saraksts.

Ja administrātors vēlās piekļūt koncerta numuru sarakstam, viņam jāpiespiež poga "Numuri" koncertu saraksta labajā pusē. Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar numuru sarakstu (skat. 58.att.).

Numuri koncertam: Ziedu dārzi									
Pievienot numuru									
Numura Nosaukums	Kārtas Numurs	Garums	Mūzika	Horeogrāfija	Izpildītāju Skaitis	Kostīms	Audzēkņi	Pedagogi	Darbības
Ziedu valsis	1	2:30	Ziedu simfonija	klasiskā deja	2	Gaišā kleita	Izpildītāji	Horeogrāfi	Dzēst
Pavasara ziedi	2	3:10	Pavasara ziedi	moderna deja	4	Bērnu svārki	Izpildītāji	Horeogrāfi	Dzēst
Sapņu pasaule	3	3:00	Sapņu dziesma	contemporary	5	Tumšā pačka	Izpildītāji	Horeogrāfi	Dzēst
Rudens lapa	4	3:15	Rudens sonāte	jazz	2	Bērnu svārki	Izpildītāji	Horeogrāfi	Dzēst
Saules deja	5	3:00	Saules rīts	moderna deja	2	Tumšā pačka	Izpildītāji	Horeogrāfi	Dzēst
Kalnu ezeri	6	3:05	Kalnu ezeri	folk	3	Garas bikses	Izpildītāji	Horeogrāfi	Dzēst
Eksportēt numuru sarakstu Eksportēt kostīmu sarakstu									

58.att. Koncerta numura saraksts

Lai dzēst numurus no koncerta, administrātoram jāpiespiež poga "Dzēst", kas atrodas saraksta labajā pusē. Pēc pogas nospiešanas, numurs tiks izdzēsts no koncerta.

Numura pievienošana koncertam.

Ja administrātors vēlās pievienot numuru koncertam, viņam zem numura nosaukuma jāizvēlās poga "Pievienot numuru". Pēc pogas piespiešanas administrātors tiks pāradresēts lapā ar numuru sarakstu, kur lai pievienot numuru koncertam ir jāpiespiež poga "Pievienot" (skat. 59.att.).

Pievienot Numuru Koncertam						
Kārtas Numurs:						
Numura Nosaukums	Kārtas Numurs	Garums	Mūzika	Horeogrāfija	Izpildītāju Skaitis	Pievienošana
Mēness serenāde		2:45	Mēness gaismā	ballets	6	Pievienot
Jūras vējš		2:50	Jūras skaņas	contemporary	4	Pievienot
Ziemas brīnums		2:40	Ziemas pasaka	klasiskā deja	3	Pievienot
Dzīvības upe		2:35	Dzīvības simfonija	jazz	4	Pievienot

59.att. Numuru saraksts pievienošanai koncertam

Pēc kārtas numura ievades un pievienošanas pogas piespiešanas, numurs tiks pievienots koncertam un tas pazudīs no pieejamo audzēkņu saraksta.

Koncerta numuru saraksta eksportēšana.

Lai eksportētu koncerta numuru sarakstu kā PDF failu, ir jāpiespiež poga "Eksportēt numuru sarakstu", kas atrodas zem koncerta numuru saraksta (skat. 60.att.).

Eksportēt numuru sarakstu

60.att. Koncerta numuru saraksta eksporta poga

Pēc pogas uzspiešanas PDF fails tiks automātiski lejupielādēts uz ierīces, to varēs atrast sadaļā "Downloads". Kad lietotājs atvērs failu sev ērtajā veidā, viņš varēs iepazīties ar koncerta numuru sarakstu (skat. 61.att.).

 Grande Ballet Studio					
Ziedu dārzi					
Datums: 2024-02-17					
Vieta: VEF kultūras pils					
Ilgums: 2 stundas					
Numuru skaits: 6					
Kārtas numurs	Nosaukums	Izpildītāji	Garums	Horeogrāfija	Pedagogi
1	Ziedu valsis	Eva Kalniņa (P4) Alise Silīņa (P1)	2:30	klasiskā deja	Iļze Krauze
2	Pavasara ziedi	Iļze Vētra (P2) Inese Ābele (P4) Maija Liepa (P1) Elza Ozoliņa (P4)	3:10	moderna deja	Iļze Krauze
3	Sapņu pasaule	Rūta Gulbe (P2) Līga Dreimane (P5) Eva Kalniņa (P4) Vilma Kalniņa (P3) Elza Ozoliņa (P4)	3:00	contemporary	Iļze Krauze
4	Rudens lapa	Vilma Kalniņa (P3) Dace Birzniece (P3)	3:15	jazz	Māris Jansons
5	Saules deja	Elza Ozoliņa (P4) Līga Kalniņa (P1)	3:00	moderna deja	Edgars Grīns
6	Kalnu ezeri	Jānis Bērziņš (P1) Rūdolfs Liepiņš (P4) Raitis Zariņš (P4)	3:05	folk	Edgars Grīns

61.att. Koncerta numuru saraksta PDF Fails

Turpmāk šo failu varēs izprintēt vai kopīgot gadījumos, kad ir nepieciešams izmantot koncerta numuru sarakstu.

Koncerta kostīmu saraksta eksportēšana.

Lai eksportētu koncerta kostīmu sarakstu kā PDF failu, ir jāpiespiež poga "Eksportēt kostīmu sarakstu", kas atrodas zem koncerta numuru saraksta (skat. 62.att.).

Eksportēt kostīmu sarakstu

62.att. Koncerta kostīmu saraksta eksporta poga

Pēc pogas uzspiešanas PDF fails tiks automātiski lejupielādēts uz ierīces, to varēs atrast sadaļā "Downloads". Kad lietotājs atvērs failu sev ērtajā veidā, viņš varēs iepazīties ar koncerta kostīmu sarakstu (skat. 63.att.).

 Grande Ballet Studio			
Ziedu dārzi			
Datums: 2024-02-17			
Vieta: VEF kultūras pils			
Ilgums: 2 stundas			
Izpildītāju skaits: 18			
Numura nosaukums	Izpildītāju skaits	Kostīma nosaukums	Kostīma attēls
Ziedu valsis	2	Gaišā kleita	
Pavasara ziedi	4	Bērnu svārki	
Sapņu pasaule	5	Tumšā pačka	
Rudens lapa	2	Bērnu svārki	
Saules deja	2	Tumšā pačka	

63.att. Koncerta kostīmu saraksta PDF Fails

Turpmāk šo failu varēs izprintēt vai kopīgot gadījumos, kad ir nepieciešams izmantot koncerta kostīmu sarakstu.

Informācija par lietotāju.

Uzklikšķinot uz lietotāja vārda un uzvārda jebkurā lietotāju sarakstā administrātors tiks pāradresēts uz lapu, kur viņam būs pieejama visa informācija par konkrēto lietotāju (skat. 64.att.).

Audzēkņa informācija

Vārds un Uzvārds	Personas Kods	Tālruņa numurs	E-pasts	Dzimšanas datums	Vecums
Eva Kalniņa	150906-27861	+37112345601	evakalnina@gmail.com	2006-09-15	17

Grupas:

Nosaukums	Grafiks	Filiale	Adrese
P4	Pirmdiena, Trešdiena 18:00-19:00	Saharova	Latvija, Rīga, Pļavnieki, Andreja Saharova iela, 35, LV-1082

Pedagogi:

Vārds un uzvārds	Tālrunis	Epasts	Dzimšanas diena
Ilze Krauze	+37198456774	ilzekrauze@gmail.com	1975-07-01

Vecāki:

Vārds un uzvārds	Tālrunis	Epasts
Ilze Kalniņa	+37122345601	ilzekalnina@gmail.com

64.att. Lietotāja informācija

Tādā veidā administrātors var piekļūt visai viņam nepieciešamai informācijai par konkrēto audzēkni, pedagogu vai vecāku.

Pedagoga funkcionalitāte, pedagoga informācija.

Veiksmīgās autentifikācijas gadījumā lietotājs ar pedagoga lomu tiek pāradresēts uz pedagoga funkcionalitātes lapu, kur augšējā lapas daļā ir pieejama informācija par viņu un viņa grupām (skat. 65.att.).

Pedagoga informācija

Vārds un Uzvārds	Tālruņa numurs	E-pasts	Kopējais nodarbību skaits nedēļā	Kopējais grupu skaits
Ilze Krauze	+37198456774	ilzekrauze@gmail.com	2	1

Grupas

Nosaukums	Grafiks	Nodarbību skaits nedēļā	Filiale	Adrese	Dalībnieki
P4	Pirmdiena, Trešdiena 18:00-19:00	2	Saharova	Latvija, Rīga, Pļavnieki, Andreja Saharova iela, 35, LV-1082	Saraksts

65.att. Pedagoga informācija

Pedagoga informācija un funkcionalitāte atrodas vienā lapā, tapēc lai piekļūtu citām sadaļām lapa jātin uz leju.

Pedagoga grupas saraksts.

Lai piekļūtu sev piešķirto grupu dalībnieku sarakstiem, pedagoga grupu tabulā jāizvēlās vēlamā grupa un jāpiespiež poga "Saraksts", kas atrodas tabulas labajā pusē (skat. 66.att.).








Eksportēt sarakstu			
Loma	Vārds un uzvārds	Tālrunis numurs	E-pasts
audzēknis	Eva Kalniņa	+3712345601	evakalnina@gmail.com
audzēknis	Artis Bērziņš	+3712345602	artisberzins@gmail.com
audzēknis	Elza Ozoliņa	+3712345603	elzaozolina@gmail.com
audzēknis	Rūdolfs Liepiņš	+3712345604	rudolfsiepins@gmail.com
audzēknis	Guna Klavina	+3712345605	gunaklavina@gmail.com
audzēknis	Raitis Zariņš	+3712345606	raitiszarins@gmail.com
audzēknis	Inese Ābele	+3712345607	ineseabele@gmail.com
pedagogs	Ilze Krauze	+37198456774	ilzekrauze@gmail.com

66.att. Pedagoga grupas informācija

Pēc pogas piespiešanas pedagogs tiks pārdresēts uz lapu, kurā varēs redzēt grupas dalībnieku sarakstu, kā arī eksportēt to PDF formātā analogiski ar to, kā to var veikt administrātors.

Pedagoga noliktavas tērpu saraksts.

Līdzīgi administrātoram pedagogam ir iespēja arī piekļūt noliktavā esošo tērpu sarakstam, tos izdalīt un savākt, kā arī eksportēt noliktavā esošo tērpu saraksta PDF failu (skat. 67.att.).

Noliktavas tērpi					
Eksportēt sarakstu					
Nosaukums	Izmērs	Krāsa	Attēls	Izdalīšana	Savākšana
Gaišā kleita	168	Gaiši rozā		Izdalīt	Savākt
Tumšā pačoka	160	Melns		Izdalīt	Savākt
Bērnu tilla svārki	xs	Spilgti rozā		Izdalīt	Savākt
Bērnu svārki	120	Tumši zils		Izdalīt	Savākt
Garas bikses	M	Melns		Izdalīt	Savākt
Spilgts kreklis	S	Sarkans		Izdalīt	Savākt
Cepure	M	Sarkans		Izdalīt	Savākt

67.att. Pedagoga noliktavā pieejamo tērpu tabula

Šo funkciju veikšanai pedagogam ir jāizmanto atbilstošas pogas, kā tas tika aprakstīts administrātoru ceļvedī.

Pedagoga koncertu saraksts.

Zem noliktavā esošo tērpu tabulas pedagogs var redzēt sarakstu ar koncertiem, kuros tiks izpildīti viņa numuri un savu numuru sarakstu, kas tiks izpildīti katrā no koncertiem (skat. 68.att.).

Koncerti

Ziedu dārzi

2024-02-17, VEF kultūras pils, 2 stundas

Eksportēt numuru sarakstu

Eksportēt tērpu sarakstu

Mani koncerta numuri:

Kārtas numurs	Nosaukums	Izpildītāji	Garums	Mūzika	Horeogrāfija	Pedagogi	Kostīmi
1	Ziedu valsis	Eva Kalniņa Alise Siliņa	2:30	Ziedu simfonija	klasiskā deja	Ilze Krauze	Gaišā kleita
2	Pavasara ziedi	Ilze Vētra Inese Ābele Maija Liepa Elza Ozoliņa	3:10	Pavasara ziedi	moderna deja	Ilze Krauze	Bērnu svārki
3	Sapņu pasaule	Rūta Gulbe Līga Dreimane Eva Kalniņa Vilma Kalniņa Elza Ozoliņa	3:00	Sapņu dziesma	contemporary	Ilze Krauze	Tumšā pačka

68.att. Pedagoga koncertu numuri

Visu koncertu informācija seko viena aiz otras, zem katra koncerta ir norādīti tie numuri, kuriem noteiktais pedagogs ir horeogrāfs.

Pedagoga koncerta numuru saraksta eksportēšana.


Lai eksportētu koncerta numuru sarakstu kā PDF failu, ir jāpiespiež poga "Eksportēt numuru sarakstu", kas atrodas virs pedagoga koncerta numuru saraksta (skat. 69.att.).

Eksportēt numuru sarakstu

69.att. Koncerta numuru saraksta eksporta poga

Pēc pogas uzspiešanas PDF fails tiks automātiski lejupielādēts uz ierīces, to varēs atrast sadaļā "Downloads".

Kad pedagogs atvērs failu sev ērtajā veidā, viņš varēs iepazīties ar savu koncerta numuru sarakstu (skat. 70.att.).



Grande Ballet Studio

Ilze Krauze - Ziedu dārzi

Datums: 2024-02-17

Vieta: VEF kultūras pils

Ilgums: 2 stundas

Kārtas numurs	Nosaukums	Izpildītājs	Garums	Mūzika	Horeogrāfija
1	Ziedu valsis	Eva Kalniņa Alise Siliņa	2:30	Ziedu simfonija	klasiskā deja
2	Pavasara ziedi	Ilze Vētra Inese Ābele Maija Liepa Elza Ozoliņa	3:10	Pavasara ziedi	moderna deja
3	Sapņu pasaule	Rūta Gulbe Līga Dreimane Eva Kalniņa Vilma Kalniņa Elza Ozoliņa	3:00	Sapņu dziesma	contemporary

70.att. Pedagoga koncerta numuru saraksta PDF Fails

Turpmāk šo failu varēs izprintēt vai kopīgot gadījumos, kad ir nepieciešams izmantot konkrēta pedagoga koncerta numuru sarakstu.





Pedagoga koncerta kostīmu saraksta eksportēšana.

Lai eksportētu pedagoga koncerta kostīmu sarakstu kā PDF failu, ir jāpiespiež poga "Eksportēt tērpu sarakstu", kas atrodas virs koncerta numuru saraksta (skat. 71.att.).

Eksportēt tērpu sarakstu

71.att. Pedagoga koncerta kostīmu saraksta eksporta poga

Pēc pogas uzspiešanas PDF fails tiks automātiski lejupielādēts uz ierīces, to varēs atrast sadaļā "Downloads". Kad pedagogs atvērs failu sev ērtajā veidā, viņš varēs iepazīties ar savu koncerta kostīmu sarakstu (skat. 72.att.).

			
Ilze Krauze - Ziedu dārzi			
Datums: 2024-02-17			
Vieta: VEF kultūras pils			
Ilgums: 2 stundas			
Numura nosaukums	Izpildītāju skaits	Kostīma nosaukums	Kostīma attēls
Ziedu valsis	2	Gaišā kleita	
Pavasara ziedi	4	Bērnu svārki	
Sapņu pasaule	5	Tumšā pačoka	

72.att. Pedagoga koncerta kostīmu saraksta PDF Fails

Turpmāk šo failu varēs izprintēt vai kopīgrot gadījumos, kad ir nepieciešams izmantot konkrēta pedagoga koncerta kostīmu sarakstu.

Pedagoga datu rediģēšana.

Pedagoga datu rediģēšanai ir jāizmanto lietotāja datu atjaunošanas forma, kas atrodas pedagoga lapas apakšējā daļā (skat. 73.att.).

Lietotāja datu atjaunošana
Vārds:

Uzvārds:

E-pasts:

Parole:


Tālrunis:

73.att. Lietotāja datu rediģēšanas forma

Lai saglabātu veiktās izmaiņas, lietotājam jāpiespiež poga "Saglabāt", lai rediģēta lietotāja dati tiktu atjaunoti.

Audzēkņa funkcionalitāte, audzēkņa informācija.

Veiksmīgās autentifikācijas gadījumā lietotājs ar audzēkņa lomu tiek pāradresēts uz audzēkņa funkcionalitātes lapu, kur augšējā lapas daļā ir pieejama informācija par viņu, viņu grupām, pedagogiem, vecākiem un kostīmiem (skat. 74.att.).

Vārds un Uzvārds	Personas Kods	Tālruna numurs	E-pasts	Dzimšanas datums	Vecums
Eva Kalniņa	150906-27861	+37112345601	evakalnina@gmail.com	2006-09-15	17
Grupās:					
Nosaukums	Grafiks	Filiale	Adrese		
P4	Pirmdiena, Trešdiena 18:00-19:00	Saharova	Latvija, Rīga, Pļavnieki, Andreja Saharova iela, 35, LV-1082		
Pedagogi:					
Vārds un uzvārds	Tālrunis	Epasts		Dzimšanas diena	
Ilze Krauze	+37198456774	ilzekrauze@gmail.com		1975-07-01	
Vecāki:					
Vārds un uzvārds	Tālrunis	Epasts			
Ilze Kalniņa	+37122345601	ilzekalnina@gmail.com			
Kostīmi:					
Nosaukums	Izmērs	Krāsa	Attēls	Izsniegšanas Datums	
Gaišā kleita	168	Gaiši rozā		2024-06-02	

74.att. Lietotāja informācija

Audzēkņa informācija un funkcionalitāte atrodas vienā lapā, tapēc lai piekļūtu citām sadaļām lapa jātin uz leju.

Audzēkņa koncertu saraksts.

Zem savu tērpu tabulas audzēknis var redzēt sarakstu ar koncertiem, kuros tiks izpildīti viņa numuri un savu numuru sarakstu, kas tiks izpildīti katrā no koncertiem (skat. 75.att.).

Koncerti, kuros piedalos

Ziedu dārzi

2024-02-17, VEF kultūras pils, 2 stundas

Eksportēt tērpu sarakstu

Mani koncerta numuri:

Kārtas numurs	Nosaukums	Garums	Mūzika	Horeogrāfija	Kostīms
1	Ziedu valsis	2:30	Ziedu simfonija	klasiskā deja	Gaišā kleita
3	Sapņu pasaule	3:00	Sapņu dziesma	contemporary	Tumšā pačoka

Ziemas svētki

2024-12-20, Dzintaru koncertzāle, 1,5 stundas

Eksportēt tērpu sarakstu

Mani koncerta numuri:

Kārtas numurs	Nosaukums	Garums	Mūzika	Horeogrāfija	Kostīms
4	Ziedu valsis	2:30	Ziedu simfonija	klasiskā deja	Gaišā kleita
7	Mēness serenāde	2:45	Mēness gaismā	ballets	Gaišā kleita
3	Sapņu pasaule	3:00	Sapņu dziesma	contemporary	Tumšā pačoka

75.att. Audzēkņa koncertu numuri

Visu koncertu informācija seko viena aiz otras, zem katra koncerta ir norādīti tie numuri, kuros piedalās konkrētais audzēknis.

Audzēkņa koncerta kostīmu saraksta eksportēšana.



Lai eksportētu audzēkņa koncerta kostīmu sarakstu kā PDF failu, ir jāpiespiež poga "Eksportēt tērpu sarakstu", kas atrodas virs koncerta numuru saraksta (skat. 76.att.).

Eksportēt tērpu sarakstu

76.att. Audzēkņa koncerta kostīmu saraksta eksporta poga

Pēc pogas uzspiešanas PDF fails tiks automātiski lejupielādēts uz ierīces, to varēs atrast sadaļā "Downloads". Kad audzēknis atvērs failu sev ērtajā veidā, viņš varēs iepazīties ar savu koncerta kostīmu sarakstu.

Turpmāk šo failu varēs izprintēt vai kopīgot gadījumos, kad ir nepieciešams izmantot konkrēta pedagoga koncerta kostīmu sarakstu (skat. 77.att.).

<div>Grande Ballet Studio</div> <div>Eva Kalniņa - Ziedu dārzi</div> <div>2024-02-17, VEF kultūras pils, 2 stundas</div> <div>Numuri</div>						
Kārtas numurs	Nosaukums	Garums	Mūzika	Horeogrāfija	Kostims	Kostīma Attēls
1	Ziedu valsis	2:30	Ziedu simfonija	klasiskā deja	Gaišā kleita	
3	Sapņu pasaule	3:00	Sapņu dziesma	contemporary	Tumšā pačoka	

77.att. Audzēkņa koncerta kostīmu saraksta PDF Fails

Audzēkņa datu rediģēšana.

Audzēkņa datu rediģēšanai ir jāizmanto lietotāja datu atjaunošanas forma, kas atrodas audzēkņa lapas apakšējā daļā (skat. 78.att.).

Dati:	
Vārds:	<input type="text" value="Eva"/>
Uzvārds:	<input type="text" value="Kalniņa"/>
E-pasts:	<input type="text" value="evakalnina@gmail.com"/>
Parole:	<input type="password" value="....."/>
Tālrunis:	<input type="text" value="+37112345601"/>
	<input type="button" value="Saglabāt"/>

78.att. Lietotāja datu rediģēšanas forma

Lai saglabātu veiktās izmaiņas, lietotājam jāpiespiež poga "Saglabāt", lai rediģēta lietotāja dati tiktu atjaunoti.

Vecāka funkcionalitāte, vecāka informācija.


Veiksmīgās autentifikācijas gadījumā lietotājs ar vecāka lomu tiek pāradresēts uz vecāka funkcionalitātes lapu, kur augšējā lapas daļā ir pieejama informācija par viņa bērnu (skat. 79.att.).

Bērni:				
Vārds un uzvārds	Personas Kods	Tālrunis	E-pasts	Dzimšanas diena
Elza Ozoliņa	170908-27863	+37112345603	elzaozolina@gmail.com	2008-09-17

79.att. Vecāka informācija

Vecāka bērna informācija.

Lai iegūtu informācija par savu bērnu, vecākam jāuzklikšķina virsū vajadzīga bērna vārdam un uzvārdam. Tājā gadījumā vecāks tiks pāradresēts uz bērna informācijas lapu (skat. 80.att.).

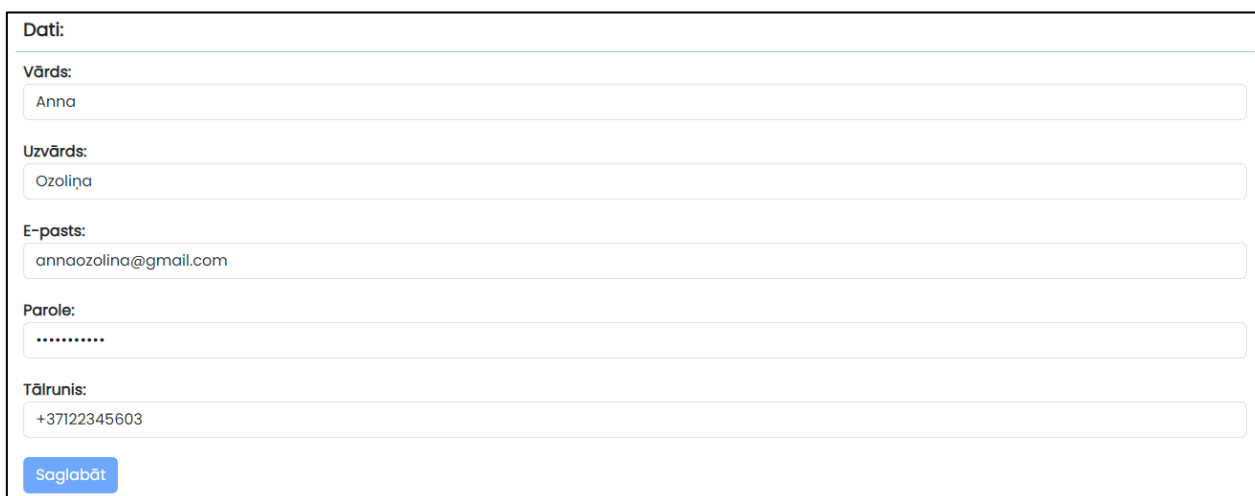
Audzēkņa informācija					
Vārds un Uzvārds	Personas Kods	Tālruņa numurs	E-pasts	Dzimšanas datums	Vecums
Elza Ozoliņa	170908-27863	+37112345603	elzaozolina@gmail.com	2008-09-17	15
Grupas:					
Nosaukums	Grafiks	Filiale	Adrese		
P4	Pirmdiena, Trešdiena 18:00-19:00	Saharova	Latvija, Rīga, Pļavnieki, Andreja Saharova iela, 35, LV-1082		
Pedagogi:					
Vārds un uzvārds	Tālrunis	Epasts	Dzimšanas diena		
Ilze Krauze	+37198456774	ilzekrauze@gmail.com	1975-07-01		
Vecāki:					
Vārds un uzvārds	Tālrunis	Epasts			
Anna Ozoliņa	+37122345603	annaozolina@gmail.com			
Kostīmi:					
Nosaukums	Izmērs	Krāsa	Attēls	Izsniegšanas Datums	
Tumšā pačka	160	Melns		2024-06-02	

80.att. Bērna informācijas lapā

Analoģiski pašam bērnam vecākam būs iespēja piekļūt informācijai par audzēkni, audzēkņa grupām, pedagogiem, vecākiem un kostīmiem, koncertu un numuru sarakstiem kā arī izdrukāt nepieciešamo koncertam tērpu sarakstu.

Vecāka datu rediģēšana.

Vecāka datu rediģēšanai ir jāizmanto lietotāja datu atjaunošanas forma, kas atrodas vecāka lapas apakšējā daļā (skat. 81.att.).



Dati:

Vārds:
Anna

Uzvārds:
Ozoliņa

E-pasts:
annaozolina@gmail.com

Parole:
.....

Tālrunis:
+37122345603

Saglabāt

81.att. Vecāka datu rediģēšanas forma

Lai saglabātu veiktās izmaiņas, lietotājam jāpiespiež poga "Saglabāt", lai rediģēta lietotāja dati tiktu atjaunoti.

Darba ar lietojumprogrammu pabeigšana

Lai pabeigtu darbu ar baletskolas kostīmu datu uzskaites sistēmu, ir nepieciešams aizvert pārlūkprogrammas logu tādā veidā, kādā to ir paredzējis darbam ar lietotni izmantojamā pārlūka izstrādātājs.

6.4. Testa piemērs

Testu piemēru izveide nodrošina iespēju identificēt un novērst kļūdas un nepilnības jau agrīnā izstrādes posmā, tādējādi uzlabojot galaprodukta kvalitāti un lietotāju pieredzi. Tālāk ir aprakstīti testa piemēri, kas tika izstrādāti un izpildīti, lai pārbaudītu baletskolas kostīmu uzskaites sistēmas funkcionalitāti.

15.tabula

Testēšanas pārskats

Nr	Prasību numurs	Prasības nosaukums	Ievaddati/sitauācības apraksts	Sagaidāmais rezultāts	Statuss
1.	1.1.	Jānodrošina, ka jaunu lietotāja kontu var pievienot tikai lietotājs ar administrātoru lomu.	Jaunu lietotāju kontu pievieno lietotājs ar administrātoru lomu.	Konts tiek veiksmīgi pievienots.	Pareizi
2.			Jaunu lietotāju kontu pievieno lietotājs ar audzēkņa lomu.	Konts nav pievienots.	Pareizi
3.			Jaunu lietotāju kontu pievieno lietotājs ar vecāka lomu.	Konts nav pievienots.	Pareizi
4.	5.3.	Ja obligātie lauki nav aizpildīti, neļaut izveidot ierakstu un izvadīt ziņojumu.	Nosaukuma lauks nav aizpildīts, pārējie lauki ir aizpildīti.	Ieraksts netika izveidots, tika izvadīts ziņojums.	Pareizi
5.			Visi lauki, izņemot izmēru, ir aizpildīti.	Ieraksts netika izveidots, tika izvadīts ziņojums.	Pareizi
6.			Visi lauki ir aizpildīti.	Ieraksts tika izveidots.	Pareizi
7.	7.3.	Lietotājam ar vecāka lomu jānodrošina piekļuve tikai viņam piesaistīto bērnu informācijai.	Lietotājam ir vecāka loma un viņam ir piesaistīts bērns.	Lietotājs var piekļūt sava bērna informācijai.	Pareizi
8.			Lietotājam ir vecāka loma un viņam nav piesaistīts bērns.	Lietotājs nevar piekļūt informācijai par citiem lietotājiem.	Pareizi
9.			Lietotājam ir audzēkņa loma.	Lietotājs var piekļūt tikai savai informācijai.	Pareizi
10.	16.1	Pedagogam jābūt iespējai piekļūt koncerta numuru sarakstam, kuros viņš piedalās, izvēloties konkrētu koncertu.	Lietotājam ir pedagoga tiesības.	Tiek attēlots koncertam atbilstošs pedagoga numuru saraksts.	Pareizi
11.			Lietotājam ir administrātoru tiesības.	Tiek attēlots visu numuru saraksts.	Pareizi
12.			Lietotājam ir audzēkņa tiesības.	Tiek attēlots tikai audzēkņa numuru saraksts.	Pareizi

Visi testa piemēri tika izpildīti, nodrošinot baletskolas kostīmu uzskaites sistēmas darbības pareizību un atbilstību noteiktajām prasībām. Veiksmīgi izpildītie testi apstiprina, ka sistēma ir gatava lietošanai, un tā spēs efektīvi apmierināt baletskolas vajadzības.

NOBEIGUMS

Kvalifikācijas darba rezultātā ir izveidota daudzfunkcionāla sistēma lietotāju kontu pārvaldībai, grupu un kostīmu administrēšanai, kā arī koncerta numuru plānošanai un izpildei. Sistēma piedāvā plašu funkcionalitāti, nodrošinot iespēju lietotājiem reģistrēties, autorizēties, rediģēt un dzēst profilus, pievienot un pārvaldīt grupas, kostīmus un koncertus, kā arī nodrošināt piekļuvi informācijai par lietotājiem, kostīmiem un numuriem. Sistēma tika izstrādāta, izmantojot programmēšanas valodas, piemēram, PHP un JavaScript, kā arī HTML un CSS priekš lietotāja saskarnes izstrādes. Datu bāzes pārvaldībai tika izmantots MySQL. Realizētās funkcijas ietver jaunu lietotāju kontu pievienošanu, lietotāju autorizāciju, profilu rediģēšanu un dzēšanu, grupu un kostīmu pievienošanu un pārvaldību, koncerta numuru plānošanu un izpildītāju pievienošanu, kā arī detalizētu atskaišu un izdruku ģenerēšanu PDF formātā.

Sistēma pašlaik var tikt izmantota izglītības iestādē, lai efektīvi pārvaldītu skolēnu, vecāku un pedagogu datus, kā arī organizētu un vadītu koncertus un to numurus. Turpmāk to varēs izmantot citās izglītības iestādēs vai organizācijās, kur nepieciešama līdzīga funkcionalitāte. Sistēmu iespējams attīstīt tālāk, pievienojot papildus funkcionalitāti, kā piemēram, sekot līdzi kostīmu skaitam noliktavā, ļaut veikt zāles rezervāciju, izmantojot šo sistēmu, pielagot pdf failu saturu savam vajadzībām, kā arī uzlabotu lietotāju saskarni un mobilo aplikāciju integrāciju. Šādi paplašinājumi ļaus vēl vairāk palielināt sistēmas efektivitāti un pielietojuma iespējas.

INFORMĀCIJAS AVOTI

1. Bootstrap apmācība - <https://www.w3schools.com/bootstrap/> - (Resurss apskatīts 31.05.2024.)
2. Bootstrap dokumentācija - <https://getbootstrap.com/docs/4.1/getting-started/introduction/> - (Resurss apskatīts 31.05.2024.)
3. DomPDF bibliotēka - <https://github.com/barryvdh/laravel-dompdf> - (Resurss apskatīts 31.05.2024.)
4. HTML - https://www.w3schools.com/html/html_colors.asp - (Resurss apskatīts 31.05.2024.)
5. Laragon dokumentācija - <https://laragon.org/docs/index.html> - (Resurss apskatīts 31.05.2024.)
6. Laravel dokumentācija - <https://laravel.com/docs/10.x> - (Resurss apskatīts 31.05.2024.)
7. Laravel pārskats - https://www.tutorialspoint.com/laravel/laravel_overview.htm - (Resurss apskatīts 31.05.2024.)
8. Sackoverflow - <https://stackoverflow.com/questions/45666167/laravel-returns-zero-for-the-primary-key-value> - (Resurss apskatīts 31.05.2024.)
9. Saišu relāciju diagramma - <https://www.lucidchart.com/pages/er-diagrams> - (Resurss apskatīts 31.05.2024.)
10. Stagestock - <https://www.stagestock.com/> - (Resurss apskatīts 31.05.2024.)
11. UML diagrammas - <https://tallyfy.com/uml-diagram/> - (Resurss apskatīts 31.05.2024.)
12. Use Case diagramma - <https://creately.com/guides/use-case-diagram-tutorial/> - (Resurss apskatīts 31.05.2024.)

PIELIKUMI

1. pielikums

Programmas pirmteksts

AddGrupasController.php

```
<?php

/**
 * Šis kontrolieris nodrošina jaunu grupu pievienošanu baletskolas kostīmu
 * uzskaites sistēmai.
 * Tas satur divas galvenās funkcijas:
 * - IndexAddGrupas(): Parāda formu jaunas grupas pievienošanai.
 * - GrupasDataInsert(): Apstrādā jaunās grupas datus un ievieto tos
 * datubāzē.
 *
 * GrupasDataInsert funkcija apstrādā ievades datus no pieprasījuma, validē
 * tos un ievieto
 * Grupas_tabula tabulā. Pēc tam tiek atgriezts paziņojums par veiksmīgu vai
 * neveiksmīgu ievietošanu,
 * atkarībā no procesa iznākuma.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: Grupas_tabula
 * Izmantotie skati: addGrupas
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Grupas_tabula;

class AddGrupasController extends Controller
{
    /**
     * Parāda formu jaunas grupas pievienošanai.
     */
    function IndexAddGrupas()
    {
        return view("addGrupas");
    }

    /**
     * Apstrādā jaunās grupas datus un ievieto tos datubāzē.
     */
    function GrupasDataInsert(Request $request)
    {
        $GrupasNosaukums = $request->input("GrupasNosaukums");
        $Grafiks = $request->input("Grafiks");
        $Filiale = $request->input("Filiale");

        $isGrupasInsertSuccess = Grupas_tabula::insert([
            "GrupasNosaukums" => $request->input("GrupasNosaukums"),
            "Grafiks" => $request->input("Grafiks"),
            "Filiale" => $request->input("Filiale"),
        ]);

        if ($isGrupasInsertSuccess) {
```

```
        echo '<script>alert("Jauna grupa veiksmīgi pievienota
;))");</script>';
    } else {
        echo '<script>alert("Jauna grupa netika pievienota
:(");</script>';
    }

    return view("addGrupas");
}
}
```

AddNumursController.php

```
<?php
```

```
/**
 * Šis kontrolieris nodrošina jaunu numuru pievienošanu baletskolas kostīmu
 uzskaites sistēmai.
 * Tas satur divas galvenās funkcijas:
 * - IndexAddNumurs(): Parāda formu jauna numura pievienošanai un nodrošina
 pieejamos kostīmus.
 * - NumursDataInsert(): Apstrādā jaunā numura datus un ievieto tos datubāzē.
 *
 * NumursDataInsert funkcija apstrādā ievades datus no pieprasījuma, validē
 tos un ievieto
 * Numurs_tabula tabulā. Pēc tam tiek atgriezts paziņojums par veiksmīgu vai
 neveiksmīgu ievietošanu,
 * atkarībā no procesa iznākuma.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: Numurs_tabula, Terpi_tabula
 * Izmantotie skati: addNumurs
 */
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
use App\Models\Numurs_tabula;
use App\Models\Terpi_tabula;
```

```
class AddNumursController extends Controller
{
```

```
    /**
     * Parāda formu jauna numura pievienošanai un nodrošina pieejamos
     kostīmus.
     *
     * @return \Illuminate\View\View
     */
```

```
    function IndexAddNumurs()
```

```
    {
        $KostimiData = Terpi_tabula::all();
        return view('addNumurs', ['KostimiData' => $KostimiData]);
    }
```

```
    /**
```

```
     * Apstrādā jaunā numura datus un ievieto tos datubāzē.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\View\View
     */
```

```
    function NumursDataInsert(Request $request)
    {
```

```
        $NumuraNosaukums = $request->input("NumuraNosaukums");
        $Garums = $request->input("Garums");
        $Muzika = $request->input("Muzika");
        $Horeografija = $request->input("Horeografija");
        $IzpilditajuSkaitis = $request->input("IzpilditajuSkaitis");
        $KostimiIDnumurs = $request->input("KostimiIDnumurs");
```

```
        $isNumursInsertSuccess = Numurs_tabula::insert([
            "NumuraNosaukums" => $request->input("NumuraNosaukums"),
            "Garums" => $request->input("Garums"),
```

```

        "Muzika" => $request->input("Muzika"),
        "Horeografija" => $request->input("Horeografija"),
        "IzpilditajuSkaitis" => $request->input("IzpilditajuSkaitis"),
        "KostimiIDnumurs" => $request->input("KostimiIDnumurs"),
    ]);

    if ($isNumursInsertSuccess) {
        echo '<script>alert("Jauns numurs veiksmīgi pievienots
;))");</script>';
    } else {
        echo '<script>alert("Jauns numurs netika pievienots
:(");</script>';
    }

    $KostimiData = Terpi_tabula::all();
    return view('addNumurs', ['KostimiData' => $KostimiData]);
}
}

```


AddTerpiController.php

```
<?php
/**
 * Šis kontrolieris nodrošina jaunu kostīmu pievienošanu baletskolas kostīmu
 uzskaites sistēmai.
 * Tas satur divas galvenās funkcijas:
 * - IndexAddTerpi(): Parāda formu jauna kostīma pievienošanai.
 * - TerpiDataInsert(): Apstrādā jaunā kostīma datus un ievieto tos datubāzē.
 *
 * TerpiDataInsert funkcija apstrādā ievades datus no pieprasījuma, validē
 tos un ievieto
 * Terpi_tabula tabulā. Ja tiek pievienots attēls, tas tiek konvertēts uz
 base64 formātu un
 * saglabāts datubāzē. Pēc tam tiek atgriezts paziņojums par veiksmīgu vai
 neveiksmīgu ievietošanu,
 * atkarībā no procesa iznākuma.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotais modelis: Terpi_tabula
 * Izmantotais skats: addTerpi
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Terpi_tabula;

class AddTerpiController extends Controller
{
    /**
     * Parāda formu jauna kostīma pievienošanai.
     *
     * @return \Illuminate\View\View
     */
    function IndexAddTerpi()
    {
        return view("addTerpi");
    }

    /**
     * Apstrādā jaunā kostīma datus un ievieto tos datubāzē.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\View\View
     */
    function TerpiDataInsert(Request $request)
    {
        $Nosaukums = $request->input("Nosaukums");
        $Krasa = $request->input("Krasa");
        $Izmers = $request->input("Izmers");
        $Attels = $request->file("Attels");

        // Pārbauda vai ir augšupielādēts attēls, un konvertē to base64
        formātā
        if ($Attels) {
            $imageData = file_get_contents($Attels->getRealPath());
            $base64 = base64_encode($imageData);
        } else {
            $base64 = null;
        }
    }
}
```

```

// Ievieto datus Terpi_tabula tabulā
$isTerpiInsertSuccess = Terpi_tabula::insert([
    "Nosaukums" => $request->input("Nosaukums"),
    "Krasa" => $request->input("Krasa"),
    "Izmers" => $request->input("Izmers"),
    "Attels" => $base64,
]);

// Izvada paziņojumu par ievietošanas rezultātu
if ($isTerpiInsertSuccess) {
    echo '<script>alert("Jauns kostīms veiksmīgi pievienots
;)"</script>';
} else {
    echo '<script>alert("Jauns kostīms netika pievienots
:(")</script>';
}

return view("addTerpi");
}
}

```

AdminController.php

```
<?php

/**
 * Šis kontrolieris nodrošina lietotāju pievienošanu un administrēšanu
 baletskolas sistēmā.
 * Tas satur divas galvenās funkcijas:
 * - AdminIndex(): Parāda administrācijas sākumlapu.
 * - UserDataInsert(): Apstrādā jaunā lietotāja datus un ievieto tos
 datubāzē.
 *
 * UserDataInsert funkcija apstrādā ievades datus no pieprasījuma, validē tos
 un ievieto
 * Lietotajs_tabula tabulā. Pēc tam tiek atgriezts paziņojums par veiksmīgu
 vai neveiksmīgu ievietošanu,
 * atkarībā no procesa iznākuma.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotais modelis: Lietotajs_tabula
 * Izmantotais skats: admin
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Lietotajs_tabula;

class AdminController extends Controller
{
    /**
     * Parāda administrācijas sākumlapu.
     *
     * @return \Illuminate\View\View
     */
    function AdminIndex()
    {
        return view("admin");
    }

    /**
     * Apstrādā jaunā lietotāja datus un ievieto tos datubāzē.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\View\View
     */
    function UserDataInsert(Request $request)
    {
        $personasKods = $request->input("personasKods");
        $Vards = $request->input("Vards");
        $Uzvards = $request->input("Uzvards");
        $Epasts = $request->input("Epasts");
        $Parole = $request->input("Parole");
        $Talrunis = $request->input("Talrunis");
        $dzimDiena = $request->input("dzimDiena");
        $LomaID = $request->input("LomaID");

        // Ievieto datus Lietotajs_tabula tabulā
        $isInsertSuccess = Lietotajs_tabula::insert([
            "personasKods" => $request->input("personasKods"),
            "Vards" => $request->input("Vards"),
```

```

        "Uzvards" => $request->input("Uzvards"),
        "Epasts" => $request->input("Epasts"),
        "Parole" => $request->input("Parole"),
        "Talrunis" => $request->input("Talrunis"),
        "dzimDiena" => $request->input("dzimDiena"),
        "LomaID" => $request->input("LomaID"),
    ]);

    // Izvada paziņojumu par ievietošanas rezultātu
    if ($isInsertSuccess) {
        echo '<script>alert("Jauns lietotājs veiksmīgi pievienots
;");</script>';
    } else {
        echo '<script>alert("Jauns lietotājs netika pievienots
:(");</script>';
    }
    return view("admin");
}
}

```

AudzekniGrupasController.php

```
<?php

/**
 * Šis kontrolieris nodrošina audzēkņu pievienošanu un dzēšanu no grupām
 baletskolas sistēmā.
 * Tas satur četras galvenās funkcijas:
 * - AudzekniGrupasIndex(): Parāda audzēkņu un pedagogu sarakstu grupai.
 * - PievAudzAudzekniGrupasIndex(): Parāda audzēkņu sarakstu, kuri vēl nav
 pievienoti grupai.
 * - AudzekniGrupasDataInsert(): Pievieno audzēkni grupai un atjauno
 sarakstu.
 * - AudzekniGrupasDataDelete(): Izdzēš audzēkni no grupas un atjauno
 sarakstu.
 *
 * AudzekniGrupasDataInsert un AudzekniGrupasDataDelete funkcijas apstrādā
 pieprasījumus,
 * veic nepieciešamās datubāzes operācijas un atgriež attiecīgos skatus ar
 atjaunoto informāciju.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: AudzeknisGrupa_tabula, Grupas_tabula, Lietotajs_tabula,
 Lomas_tabula
 * Izmantotie skati: audzekniGrupas, pievAudzAudzekniGrupas
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\AudzeknisGrupa_tabula;
use App\Models\Grupas_tabula;
use App\Models\Lietotajs_tabula;
use App\Models\Lomas_tabula;

class AudzekniGrupasController extends Controller
{
    /**
     * Parāda audzēkņu un pedagogu sarakstu grupai.
     *
     * @param string $GrupasNosaukums
     * @return \Illuminate\View\View
     */
    function AudzekniGrupasIndex($GrupasNosaukums)
    {
        $LietotajsData = Lietotajs_tabula::whereIn("LomaID", [1, 3])
            ->where(function ($query) use ($GrupasNosaukums) {
                $query
                    ->whereExists(function ($subquery) use ($GrupasNosaukums)
                    {
                        $subquery
                            ->select("*")
                            ->from("audzeknisgrupa")
                            ->whereRaw(
                                "AudzeknaGrupasPersonasKods = personasKods"
                            )
                            ->where(
                                "GrupasAudzeknaNosaukums",
                                $GrupasNosaukums
                            );
                    });
            });
    }
}
```

```

        ->orWhereExists(function ($subquery) use (
            $GrupasNosaukums
        ) {
            $subquery
                ->select("*")
                ->from("pedagogsgrupa")
                ->whereRaw("PedagogaPersonasKods = personasKods")
                ->where(
                    "GrupasPedagogaNosaukums",
                    $GrupasNosaukums
                );
        });
    });
    ->get();

    $LomasData = Lomas_tabula::all();
    $GrupasData = Grupas_tabula::all();

    return view("audzekniGrupas", [
        "LietotajsData" => $LietotajsData,
        "LomasData" => $LomasData,
        "GrupasNosaukums" => $GrupasNosaukums,
        "GrupasData" => $GrupasData,
    ]);
}

/**
 * Parāda audzēkņu sarakstu, kuri vēl nav pievienoti grupai.
 *
 * @param string $GrupasNosaukums
 * @return \Illuminate\View\View
 */
function PievAudzAudzekniGrupasIndex($GrupasNosaukums)
{
    $LietotajsData = Lietotajs_tabula::where("LomaID", 1)
        ->whereNotExists(function ($query) use ($GrupasNosaukums) {
            $query
                ->select("*")
                ->from("audzeknisgrupa")
                ->whereRaw("AudzeknaGrupasPersonasKods = PersonasKods")
                ->where("GrupasAudzeknaNosaukums", $GrupasNosaukums);
        })
        ->get();

    $LomasData = Lomas_tabula::all();
    $GrupasData = Grupas_tabula::all();

    return view("pievAudzAudzekniGrupas", [
        "GrupasNosaukums" => $GrupasNosaukums,
        "LietotajsData" => $LietotajsData,
        "LomasData" => $LomasData,
    ]);
}

/**
 * Pievieno audzēkni grupai un atjauno sarakstu.
 *
 * @param string $GrupasNosaukums
 * @param string $personasKods
 * @return \Illuminate\View\View
 */

```

```

function AudzekniGrupasDataInsert($GrupasNosaukums, $personasKods)
{
    $date = date("Y-m-d");
    $isAudzekniGrupasInsertSuccess = AudzeknisGrupa_tabula::insert([
        "PienDatums" => $date,
        "AudzeknaGrupasPersonasKods" => $personasKods,
        "GrupasAudzeknaNosaukums" => $GrupasNosaukums,
    ]);

    if ($isAudzekniGrupasInsertSuccess) {
        echo '<script>alert("Audzēknis tika pievienots grupai
;)"</script>';
    } else {
        echo '<script>alert("Audzēknis netika pievienots grupai
:(")</script>';
    }

    $LietotajsData = Lietotajs_tabula::whereIn("LomaID", [1, 3])
        ->where(function ($query) use ($GrupasNosaukums) {
            $query
                ->whereExists(function ($subquery) use ($GrupasNosaukums)
{
                    $subquery
                        ->select("*")
                        ->from("audzeknisgrupa")
                        ->whereRaw(
                            "AudzeknaGrupasPersonasKods = personasKods"
                        )
                        ->where(
                            "GrupasAudzeknaNosaukums",
                            $GrupasNosaukums
                        );
                })
                ->orWhereExists(function ($subquery) use (
                    $GrupasNosaukums
                ) {
                    $subquery
                        ->select("*")
                        ->from("pedagogsgrupa")
                        ->whereRaw("PedagogaPersonasKods = personasKods")
                        ->where(
                            "GrupasPedagogaNosaukums",
                            $GrupasNosaukums
                        );
                });
            });
        ->get();

    $LomasData = Lomas_tabula::all();
    $GrupasData = Grupas_tabula::all();

    return view("audzekniGrupas", [
        "LietotajsData" => $LietotajsData,
        "LomasData" => $LomasData,
        "GrupasNosaukums" => $GrupasNosaukums,
        "GrupasData" => $GrupasData,
    ]);
}

/**
 * Izdzēš audzēkni no grupas un atjauno sarakstu.

```

```

*
* @param string $GrupasNosaukums
* @param string $personasKods
* @return \Illuminate\View\View
*/
function AudzekniGrupasDataDelete($GrupasNosaukums, $personasKods)
{
    $isAudzekniGrupasDeleteSuccess = AudzeknisGrupa_tabula::where([
        ["GrupasAudzeknaNosaukums", "=", $GrupasNosaukums],
        ["AudzeknaGrupasPersonasKods", "=", $personasKods],
    ])->delete();

    if ($isAudzekniGrupasDeleteSuccess) {
        echo '<script>alert("Audzēknis tika izdzēsts no
grupas");</script>';
    } else {
        echo '<script>alert("Audzēknis netika izdzēsts no
grupas");</script>';
    }
    $LietotajsData = Lietotajs_tabula::where("LomaID", 1)
        ->whereNotExists(function ($query) use ($GrupasNosaukums) {
            $query
                ->select("*")
                ->from("audzeknisgrupa")
                ->whereRaw("AudzeknaGrupasPersonasKods = PersonasKods")
                ->where("GrupasAudzeknaNosaukums", $GrupasNosaukums);
        })
        ->get();

    $LomasData = Lomas_tabula::all();
    $GrupasData = Grupas_tabula::all();

    return view("pievAudzAudzekniGrupas", [
        "GrupasNosaukums" => $GrupasNosaukums,
        "LietotajsData" => $LietotajsData,
        "LomasData" => $LomasData,
    ]);
}
}

```


AudzekniKostimiController.php

```
<?php

/**
 * Šis kontrolieris nodrošina audzēkņu kostīmu piešķiršanu un savākšanu
 baletskolas sistēmā.
 * Tas satur piecas galvenās funkcijas:
 * - AudzekniKostimiIndex(): Parāda audzēkņu sarakstu, kuriem nav piešķirts
 kostīms.
 * - AudzekniKostimiDataInsert(): Pievieno kostīmu audzēknim un atjauno
 sarakstu.
 * - AudzekniKostimiSavaktIndex(): Parāda audzēkņu sarakstu, kuriem ir
 piešķirts kostīms.
 * - AudzekniKostimiDataDelete(): Noņem kostīmu no audzēkņa un atjauno
 sarakstu.
 * - AudzekniKostimiGrupaIndex(): Parāda grupu un audzēkņu sarakstu ar
 piešķirtajiem kostīmiem.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: AudzekniKostimi_tabula, Terpi_tabula, Lietotajs_tabula,
 Lomas_tabula, AudzeknisGrupa_tabula, Grupas_tabula
 * Izmantotie skati: audzekniKostimi, audzekniKostimiSavakt,
 audzekniKostimiGrupas
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Models\AudzekniKostimi_tabula;
use App\Models\Terpi_tabula;
use App\Models\Lietotajs_tabula;
use App\Models\Lomas_tabula;
use App\Models\AudzeknisGrupa_tabula;
use App\Models\Grupas_tabula;

class AudzekniKostimiController extends Controller
{
    /**
     * Parāda audzēkņu sarakstu, kuriem nav piešķirts kostīms.
     *
     * @param int $KostimiID
     * @return \Illuminate\View\View
     */
    function AudzekniKostimiIndex($KostimiID)
    {
        $LietotajsData = Lietotajs_tabula::where("LomaID", 1)
            ->whereNotExists(function ($query) use ($KostimiID) {
                $query
                    ->select("*")
                    ->from("audzeknikostimi")
                    ->whereRaw("AudzeknaKostimaPersonasKods = PersonasKods")
                    ->where("KostimiID", $KostimiID);
            })
            ->leftJoin(
                "audzeknisgrupa",
                "lietotajs.personasKods",
                "=",
                "audzeknisgrupa.AudzeknaGrupasPersonasKods"
            )
    }
}
```

```

        ->get([
            "lietotajs.personasKods",
            "lietotajs.Vards",
            "lietotajs.Uzvards",
            "audzeknisgrupa.GrupasAudzeknaNosaukums",
        ]);
    $LomasData = Lomas_tabula::all();
    $KostimiData = Terpi_tabula::all();
    return view("audzekniKostimi", [
        "KostimiID" => $KostimiID,
        "LietotajsData" => $LietotajsData,
        "LomasData" => $LomasData,
    ]);
}

/**
 * Pievieno kostīmu audzēknim un atjauno sarakstu.
 *
 * @param int $KostimiID
 * @param string $personasKods
 * @return \Illuminate\Http\RedirectResponse
 */
function AudzekniKostimiDataInsert($KostimiID, $personasKods)
{
    $date = date("Y-m-d");

    $isAudzekniKostimiInsertSuccess = AudzekniKostimi_tabula::insert([
        "PiesDatums" => $date,
        "AudzeknaKostimaPersonasKods" => $personasKods,
        "KostimiID" => $KostimiID,
    ]);
    if ($isAudzekniKostimiInsertSuccess) {
        return back()->with(["success" => true]);
    } else {
        return back()->with(["success" => false]);
    }
}

/**
 * Parāda audzēkņu sarakstu, kuriem ir piešķirts kostīms.
 *
 * @param int $KostimiID
 * @return \Illuminate\View\View
 */
function AudzekniKostimiSavaktIndex($KostimiID)
{
    $LietotajsData = Lietotajs_tabula::where("LomaID", 1)
        ->whereExists(function ($query) use ($KostimiID) {
            $query
                ->select("*")
                ->from("audzeknikostimi")
                ->whereRaw("AudzeknaKostimaPersonasKods = PersonasKods")
                ->where("KostimiID", $KostimiID);
        })
        ->leftJoin(
            "audzeknisgrupa",
            "lietotajs.personasKods",
            "=",
            "audzeknisgrupa.AudzeknaGrupasPersonasKods"
        )
        ->get([

```

```

        "lietotajs.personasKods",
        "lietotajs.Vards",
        "lietotajs.Uzvards",
        "audzeknisgrupa.GrupasAudzeknaNosaukums",
    ]);
    $LomasData = Lomas_tabula::all();
    $KostimiData = Terpi_tabula::all();
    return view("audzekniKostimiSavakt", [
        "KostimiID" => $KostimiID,
        "LietotajsData" => $LietotajsData,
        "LomasData" => $LomasData,
    ]);
}

/**
 * Noņem kostīmu no audzēkņa un atjauno sarakstu.
 *
 * @param int $KostimiID
 * @param string $personasKods
 * @return \Illuminate\Http\RedirectResponse
 */
function AudzekniKostimiDataDelete($KostimiID, $personasKods)
{
    $isAudzekniKostimiDeleteSuccess = AudzekniKostimi_tabula::where([
        ["KostimiID", "=", $KostimiID],
        ["AudzeknaKostimaPersonasKods", "=", $personasKods],
    ])->delete();

    if ($isAudzekniKostimiDeleteSuccess) {
        return back()->with(["success" => true]);
    } else {
        return back()->with(["success" => false]);
    }
}

/**
 * Parāda grupu un audzēkņu sarakstu ar piešķirtajiem kostīmiem.
 *
 * @param int $KostimiID
 * @return \Illuminate\View\View
 */
function AudzekniKostimiGrupaIndex($KostimiID)
{
    $GrupasData = Grupas_tabula::all();
    $KostimiData = Terpi_tabula::all();
    $audzekniKostimiData = AudzekniKostimi_tabula::all();
    return view("audzekniKostimiGrupas", [
        "KostimiID" => $KostimiID,
        "GrupasData" => $GrupasData,
        "audzekniKostimiData" => $audzekniKostimiData,
    ]);
}
}

```

AudzekniNumursController.php

```
<?php

/**
 * Šis kontrolieris nodrošina audzēkņu pievienošanu un noņemšanu no numuriem
 * baletskolas sistēmā.
 * Tas satur četras galvenās funkcijas:
 * - AudzekniNumursIndex(): Parāda audzēkņu sarakstu, kuri ir pievienoti
 *   konkrētam numuram.
 * - PievAudzAudzekniNumursIndex(): Parāda audzēkņu sarakstu, kuri nav
 *   pievienoti konkrētam numuram.
 * - AudzekniNumursDataInsert(): Pievieno audzēkni numuram un atjauno
 *   sarakstu.
 * - AudzekniNumursDataDelete(): Noņem audzēkni no numura un atjauno
 *   sarakstu.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: AudzeknisNumurs_tabula, Numurs_tabula,
 * Lietotajs_tabula, Lomas_tabula, AudzeknisGrupa_tabula
 * Izmantotie skati: audzekniNumurs, pievAudzAudzekniNumurs
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\AudzeknisNumurs_tabula;
use App\Models\Numurs_tabula;
use App\Models\Lietotajs_tabula;
use App\Models\Lomas_tabula;
use App\Models\AudzeknisGrupa_tabula;

class AudzekniNumursController extends Controller
{
    /**
     * Parāda audzēkņu sarakstu, kuri ir pievienoti konkrētam numuram.
     *
     * @param int $NumursID
     * @return \Illuminate\View\View
     */
    public function AudzekniNumursIndex($NumursID)
    {
        $LietotajsData = Lietotajs_tabula::where('LomaID', 1)
            ->whereExists(function ($subquery) use ($NumursID) {
                $subquery
                    ->select('*')
                    ->from('audzeknisnumurs')
                    ->whereRaw('AudzeknaNumuraPersonasKods = personasKods')
                    ->where('NumursIDAudzeknis', $NumursID);
            })
            ->get();

        foreach ($LietotajsData as $lietotajs) {
            $grupa =
                AudzeknisGrupa_tabula::where('AudzeknaGrupasPersonasKods', $lietotajs-
                    >personasKods)
                    ->join('grupa', 'grupa.GrupasNosaukums', '=',
                        'audzeknisgrupa.GrupasAudzeknaNosaukums')
                    ->select('grupa.GrupasNosaukums')
                    ->first();
            $lietotajs->grupa = $grupa ? $grupa->GrupasNosaukums : 'N/A';
        }
    }
}
```

```

    $LomasData = Lomas_tabula::all();
    $NumursData = Numurs_tabula::find($NumursID);

    return view('audzekniNumurs', [
        'LietotajsData' => $LietotajsData,
        'LomasData' => $LomasData,
        'NumursData' => $NumursData,
    ]);
}

/**
 * Parāda audzēkņu sarakstu, kuri nav pievienoti konkrētam numuram.
 *
 * @param int $NumursID
 * @return \Illuminate\View\View
 */
public function PievAudzAudzekniNumursIndex($NumursID)
{
    $LietotajsData = Lietotajs_tabula::where('LomaID', 1)
        ->whereNotExists(function ($query) use ($NumursID) {
            $query
                ->select('*')
                ->from('audzeknisnumurs')
                ->whereRaw('AudzeknaNumuraPersonasKods = personasKods')
                ->where('NumursIDAudzeknis', $NumursID);
        })
        ->get();

    foreach ($LietotajsData as $lietotajs) {
        $grupa =
AudzeknisGrupa_tabula::where('AudzeknaGrupasPersonasKods', $lietotajs-
>personasKods)
            ->join('grupa', 'grupa.GrupasNosaukums', '=',
'audzeknisgrupa.GrupasAudzeknaNosaukums')
            ->select('grupa.GrupasNosaukums')
            ->first();
        $lietotajs->grupa = $grupa ? $grupa->GrupasNosaukums : 'N/A';
    }

    $LomasData = Lomas_tabula::all();
    $NumursData = Numurs_tabula::find($NumursID);

    return view('pievAudzAudzekniNumurs', [
        'NumursData' => $NumursData,
        'LietotajsData' => $LietotajsData,
        'LomasData' => $LomasData,
    ]);
}

/**
 * Pievieno audzēkni numuram un atjauno sarakstu.
 *
 * @param int $NumursID
 * @param string $personasKods
 * @return \Illuminate\View\View
 */
public function AudzekniNumursDataInsert($NumursID, $personasKods)
{
    $isAudzekniNumursInsertSuccess = AudzeknisNumurs_tabula::insert([
        "AudzeknaNumuraPersonasKods" => $personasKods,

```

```

        "NumursIDAudzeknis" => $NumursID,
    });

    if ($isAudzekniNumursInsertSuccess) {
        echo '<script>alert("Audzēknis tika pievienots numuram
;)"</script>';
    } else {
        echo '<script>alert("Audzēknis netika pievienots numuram
:(")</script>';
    }

    return $this->AudzekniNumursIndex($NumursID);
}

/**
 * Noņem audzēkni no numura un atjauno sarakstu.
 *
 * @param int $NumursID
 * @param string $personasKods
 * @return \Illuminate\View\View
 */
public function AudzekniNumursDataDelete($NumursID, $personasKods)
{
    $isAudzekniNumursDeleteSuccess = AudzeknisNumurs_tabula::where([
        ['NumursIDAudzeknis', '=', $NumursID],
        ['AudzeknaNumuraPersonasKods', '=', $personasKods],
    ]->delete();

    if ($isAudzekniNumursDeleteSuccess) {
        echo '<script>alert("Audzēknis tika izdzēsts no
numura")</script>';
    } else {
        echo '<script>alert("Audzēknis netika izdzēsts no
numura")</script>';
    }

    return $this->PievAudzAudzekniNumursIndex($NumursID);
}
}

```

DeleteGrupasController.php

```
<?php

/**
 * Šis kontrolieris nodrošina grupu dzēšanu baletskolas sistēmā.
 * Tas satur divas galvenās funkcijas:
 * - DeleteGrupasIndex(): Parāda lapu, kurā var apstiprināt grupas dzēšanu.
 * - DeleteGrupasData(): Dzēš norādīto grupu un atjauno grupu sarakstu.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: Grupas_tabula
 * Izmantotie skati: deleteGrupas, viewGrupas
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Grupas_tabula;

class DeleteGrupasController extends Controller
{
    /**
     * Parāda lapu, kurā var apstiprināt grupas dzēšanu.
     *
     * @param string $GrupasNosaukums
     * @return \Illuminate\View\View
     */
    function DeleteGrupasIndex($GrupasNosaukums)
    {
        return view("deleteGrupas", ["GrupasNosaukums" => $GrupasNosaukums]);
    }

    /**
     * Dzēš norādīto grupu un atjauno grupu sarakstu.
     *
     * @param string $GrupasNosaukums
     * @return \Illuminate\View\View
     */
    function DeleteGrupasData($GrupasNosaukums)
    {
        $isDeleteSuccess = Grupas_tabula::where(
            "GrupasNosaukums",
            $GrupasNosaukums
        )->delete();

        if ($isDeleteSuccess) {
            echo '<script>alert("Grupa tika izdzēsta");</script>';
        } else {
            echo '<script>alert("Grupa netika izdzēsta");</script>';
        }

        $GrupasData = Grupas_tabula::all();
        return view("viewGrupas", ["GrupasData" => $GrupasData]);
    }
}
```

KoncertiNumuriController.php

```
<?php

/**
 * Šis kontrolieris nodrošina koncertu numuru pārvaldību baletskolas sistēmā.
 * Tas satur piecas galvenās funkcijas:
 * - KoncertiNumuriIndex(): Parāda visus numurus, kas pievienoti konkrētajam
koncertam.
 * - PievNumuriKoncertamIndex(): Parāda lapu numuru pievienošanai koncertam.
 * - KoncertiNumuriDataInsert(): Pievieno jaunu numuru koncertam datubāzei.
 * - KoncertiNumuriDataDelete(): Dzēš numuru no koncerta un atjauno numuru
sarakstu.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: Koncerts_tabula, Numurs_tabula, KoncertsNumurs_tabula
 * Izmantotie skati: koncertiNumuri, pievNumuriKoncertam
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Koncerts_tabula;
use App\Models\Numurs_tabula;
use App\Models\KoncertsNumurs_tabula;

class KoncertiNumuriController extends Controller
{
    /**
     * Parāda visus numurus, kas pievienoti konkrētajam koncertam.
     */
    public function KoncertiNumuriIndex($KoncertsID)
    {
        $NumuriData = Numurs_tabula::join('koncertsnumurs',
'numurs.NumursID', '=', 'koncertsnumurs.NumursIDKoncNumurs')
->join('kostimi', 'numurs.KostimiIDnumurs', '=',
'kostimi.KostimiID')
->where('koncertsnumurs.KoncertsIDKoncNumurs', $KoncertsID)
->orderBy('koncertsnumurs.KartasNumurs')
->get(['numurs.*', 'koncertsnumurs.KartasNumurs',
'koncertsnumurs.KoncertsNumursID', 'kostimi.Nosaukums']);

        $koncerts = Koncerts_tabula::find($KoncertsID);

        return view('koncertiNumuri', [
            'NumuriData' => $NumuriData,
            'koncerts' => $koncerts,
        ]);
    }

    /**
     * Parāda lapu numuru pievienošanai koncertam.
     */
    public function PievNumuriKoncertamIndex($KoncertsID)
    {
        $NumuriData = Numurs_tabula::whereNotExists(function ($query) use
($KoncertsID) {
            $query
                ->select('*')
                ->from('koncertsnumurs')
        })
        ->select('*')
        ->from('koncertsnumurs')
        ->where('koncertsnumurs.KoncertsIDKoncNumurs', $KoncertsID)
        ->orderBy('koncertsnumurs.KartasNumurs')
        ->get();

        $koncerts = Koncerts_tabula::find($KoncertsID);

        return view('koncertiNumuri', [
            'NumuriData' => $NumuriData,
            'koncerts' => $koncerts,
        ]);
    }
}
```



```

        ->whereRaw('koncertsnumurs.NumursIDKoncNumurs =
numurs.NumursID')
        ->where('koncertsnumurs.KoncertsIDKoncNumurs', $KoncertsID);
    })->get();

    return view('pievNumuriKoncertam', [
        'KoncertsID' => $KoncertsID,
        'NumuriData' => $NumuriData,
    ]);
}

/**
 * Pievieno jaunu numuru koncertam datubāzei.
 */
public function KoncertiNumuriDataInsert(Request $request, $KoncertsID,
$NumursID)
{
    $kartasNumurs = $request->input('KartasNumurs');
    $isInsertSuccess = KoncertsNumurs_tabula::insert([
        "KoncertsIDKoncNumurs" => $KoncertsID,
        "KartasNumurs" => $kartasNumurs,
        "NumursIDKoncNumurs" => $NumursID,
    ]);

    if ($isInsertSuccess) {
        echo '<script>alert("Numurs tika pievienots koncertam
;)"</script>';
    } else {
        echo '<script>alert("Numurs netika pievienots koncertam
:(")</script>';
    }

    return redirect()->back();
}

/**
 * Dzēš numuru no koncerta un atjauno numuru sarakstu.
 */
public function KoncertiNumuriDataDelete($KoncertsNumursID)
{
    $isDeleteSuccess = KoncertsNumurs_tabula::where('KoncertsNumursID',
$KoncertsNumursID)->delete();

    if ($isDeleteSuccess) {
        echo '<script>alert("Numurs tika izdzēsts no koncerta
;)"</script>';
    } else {
        echo '<script>alert("Numurs netika izdzēsts no koncerta
:(")</script>';
    }

    return redirect()->back();
}
}
}

```

LoginController.php

```
<?php

/**
 * Šis kontrolieris nodrošina lietotāju autentifikāciju un dažādu lomu vadību
 baletskolas sistēmā.
 * Tas satur sešas galvenās funkcijas:
 * - LoginIndex(): Parāda pieteikšanās lapu.
 * - LoginPost(): Apstrādā lietotāja pieteikšanās datus un pārbauda, vai tie
 ir pareizi.
 * - AudzeknisIndex(): Parāda audzēkņa informācijas lapu, ieskaitot grupas,
 pedagogus, tērpus, vecākus, koncertus un numurus.
 * - VecaksIndex(): Parāda vecāka informācijas lapu, ieskaitot bērna datus.
 * - PedagogsIndex(): Parāda pedagoga informācijas lapu, ieskaitot grupas,
 audzēkņus, vecākus, tērpus un koncertus.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: Lietotajs_tabula, Grupas_tabula, AudzeknisGrupa_tabula,
 Terpi_tabula, BernsVecaks_tabula, Koncerts_tabula, Numurs_tabula,
 AudzeknisNumurs_tabula, PedagogNumurs_tabula, KoncertsNumurs_tabula
 * Izmantotie skati: login, admin, audzeknis, vecaks, pedagogs
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Lietotajs_tabula;
use App\Models\Grupas_tabula;
use App\Models\AudzeknisGrupa_tabula;
use App\Models\Terpi_tabula;
use App\Models\BernsVecaks_tabula;
use App\Models\Koncerts_tabula;
use App\Models\Numurs_tabula;
use App\Models\AudzeknisNumurs_tabula;
use App\Models\PedagogNumurs_tabula;
use App\Models\KoncertsNumurs_tabula;

class LoginController extends Controller
{
    /**
     * Parāda pieteikšanās lapu.
     */
    function LoginIndex()
    {
        return view("login");
    }

    /**
     * Apstrādā lietotāja pieteikšanās datus un pārbauda, vai tie ir pareizi.
     */
    function LoginPost(Request $request)
    {
        $personasKods = $request->input("personasKods");
        $parole = $request->input("Parole");

        $lietotajs = Lietotajs_tabula::where("personasKods", $personasKods)
            ->where("parole", $parole)
            ->first();

        if ($lietotajs) {
```

```

        session([
            "personasKods" => $personasKods,
            "LomaID" => $lietotajs->LomaID,
        ]);
        echo '<script>alert(" ");</script>';
        switch (session("LomaID")) {
            case 0:
                return view("admin");
                break;
            case 1:
                return $this->AudzeknisIndex($personasKods);
                break;
            case 2:
                return $this->VecaksIndex($personasKods);
                break;
            case 3:
                return $this->PedagogsIndex($personasKods);
                break;
        }
    } else {
        echo '<script>alert(" :( Pārbaudiet personas kodu un
        paroli");</script>';
        return view("login");
    }
}

/**
 * Parāda audzēkņa informācijas lapu, ieskaitot grupas, pedagogus,
 * tērpus, vecākus, koncertus un numurus.
 */
function AudzeknisIndex($personasKods)
{
    $LietotajsData = Lietotajs_tabula::where("personasKods",
    $personasKods)->first();

    $GrupasData = Grupas_tabula::join("audzeknisgrupa",
    "grupa.GrupasNosaukums", "=", "audzeknisgrupa.GrupasAudzeknaNosaukums")
    ->join("filiale", "grupa.Filiale", "=", "filiale.Filiale")
    ->where("audzeknisgrupa.AudzeknaGrupasPersonasKods",
    $personasKods)
    ->select("grupa.*", "filiale.Valsts", "filiale.Pilseta",
    "filiale.Rajons", "filiale.Iela", "filiale.Eka", "filiale.Indekss")
    ->distinct()
    ->get();

    $PedagogsData = Lietotajs_tabula::where("LomaID", 3)
    ->where(function ($query) use ($GrupasData) {
        foreach ($GrupasData as $grupa) {
            $GrupasNosaukums = $grupa->GrupasNosaukums;
            $query->orWhereExists(function ($subquery) use
            ($GrupasNosaukums) {
                $subquery
                ->select("*")
                ->from("pedagogsgrupa")
                ->whereRaw("PedagogaPersonasKods = personasKods")
                ->where("GrupasPedagogaNosaukums",
                $GrupasNosaukums);
            });
        }
    });
    ->get();
}

```

```

        $TerpiData = Terpi_tabula::join("audzeknikostimi",
"kostimi.KostimiID", "=", "audzeknikostimi.KostimiID")
        ->where("audzeknikostimi.AudzeknaKostimaPersonasKods",
$personasKods)
        ->select("kostimi.*", "audzeknikostimi.PiesDatums")
        ->get();

        $VecaksData = BernsVecaks_tabula::join("lietotajs",
"bernsvecaks.VecakaPersonasKods", "=", "lietotajs.personasKods")
        ->where("bernsvecaks.BernaPersonasKods", $personasKods)
        ->get(["lietotajs.Vards", "lietotajs.Uzvards",
"lietotajs.personasKods", "lietotajs.Talrunis", "lietotajs.Epasts",
"lietotajs.DzimDiena"]);

        $KoncertiData = Koncerts_tabula::join('koncertsnumurs',
'koncerts.KoncertsID', '=', 'koncertsnumurs.KoncertsIDKoncNumurs')
        ->join('audzeknisnumurs', 'koncertsnumurs.NumursIDKoncNumurs',
'=', 'audzeknisnumurs.NumursIDAudzeknis')
        ->where('audzeknisnumurs.AudzeknaNumuraPersonasKods',
$personasKods)
        ->select('koncerts.*')
        ->distinct()
        ->get();

        $NumuriData = Numurs_tabula::join('audzeknisnumurs',
'numurs.NumursID', '=', 'audzeknisnumurs.NumursIDAudzeknis')
        ->join('koncertsnumurs', 'numurs.NumursID', '=',
'koncertsnumurs.NumursIDKoncNumurs')
        ->join('kostimi', 'numurs.KostimiIDnumurs', '=',
'kostimi.KostimiID')
        ->where('audzeknisnumurs.AudzeknaNumuraPersonasKods',
$personasKods)
        ->select('numurs.*', 'koncertsnumurs.KartasNumurs',
'koncertsnumurs.KoncertsIDKoncNumurs', 'kostimi.Nosaukums as
KostimaNosaukums')
        ->get();

        return view("audzeknis", [
            "ld" => $LietotajsData,
            "GrupasData" => $GrupasData,
            "PedagogsData" => $PedagogsData,
            "TerpiData" => $TerpiData,
            "VecaksData" => $VecaksData,
            "KoncertiData" => $KoncertiData,
            "NumuriData" => $NumuriData,
        ]);
    }

    /**
     * Parāda vecāka informācijas lapu, ieskaitot bērna datus.
     */
    function VecaksIndex($personasKods)
    {
        $LietotajsData = Lietotajs_tabula::where(
            "personasKods",
            $personasKods
        )->get();

        $BernsData = BernsVecaks_tabula::join(
            "lietotajs",

```

```

        "bernsvecaks.BernaPersonasKods",
        "=",
        "lietotajs.personasKods"
    )
    ->where("bernsvecaks.VecakaPersonasKods", $personasKods)
    ->get([
        "lietotajs.Vards",
        "lietotajs.Uzvards",
        "lietotajs.personasKods",
        "lietotajs.Talrunis",
        "lietotajs.Epasts",
        "lietotajs.DzimDiena",
    ]);

    if (count($LietotajsData) > 0) {
        return view("vecaks", [
            "ld" => $LietotajsData[0],
            "BernsData" => $BernsData,
        ]);
    } else {
        return view("vecaks", ["ld" => null]);
    }
}

/**
 * Parāda pedagoga informācijas lapu, ieskaitot grupas, audzēkņus,
 * vecākus, tērpus un koncertus.
 */
public function PedagogIndex($personasKods)
{
    $LietotajsData = Lietotajs_tabula::where('personasKods',
    $personasKods)->first();

    $GrupasData = Grupas_tabula::join('pedagogsgrupa',
    'grupa.GrupasNosaukums', '=', 'pedagogsgrupa.GrupasPedagogaNosaukums')
    ->join('filiale', 'grupa.Filiale', '=', 'filiale.Filiale')
    ->where('pedagogsgrupa.PedagogaPersonasKods', $personasKods)
    ->select('grupa.*', 'filiale.Valsts', 'filiale.Pilseta',
    'filiale.Rajons', 'filiale.Iela', 'filiale.Eka', 'filiale.Indekss',
    'pedagogsgrupa.*')
    ->distinct()
    ->get();

    $AudzeknisData = Lietotajs_tabula::where('LomaID', 1)
    ->whereIn('personasKods', function($query) use ($GrupasData) {
        $query->select('AudzeknaGrupasPersonasKods')
        ->from('audzeknisgrupa')
        ->whereIn('GrupasAudzeknaNosaukums', $GrupasData-
    >pluck('GrupasNosaukums')));
    })
    ->get();

    $VecakiData = BernsVecaks_tabula::join('lietotajs as vecaki',
    'bernsvecaks.VecakaPersonasKods', '=', 'vecaki.personasKods')
    ->join('lietotajs as berni', 'bernsvecaks.BernaPersonasKods',
    '=', 'berni.personasKods')
    ->whereIn('berni.personasKods', $AudzeknisData-
    >pluck('personasKods'))
    ->get(['berni.Vards as BernsVards', 'berni.Uzvards as
    BernsUzvards', 'berni.personasKods as BernsPersonasKods', 'vecaki.Vards as

```

```

VecaksVards', 'vecaki.Uzvards as VecaksUzvards', 'vecaki.Talrunis as
VecaksTalrunis']]);

$TerpiData = Terpi_tabula::all();

$KoncertiData = Koncerts_tabula::all();

$KoncertiNumuri = KoncertsNumurs_tabula::with(['numurs', 'koncerts',
'numurs.audzekni', 'numurs.pedagogi'])
->whereHas('numurs.pedagogi', function($query) use
($personasKods) {
    $query->where('PedagogaNumuraPersonasKods', $personasKods);
})
->get();

$nodarbibuSkaitis = $GrupasData->sum('NodSk');
$grupuSkaitis = $GrupasData->count();

return view('pedagogs', compact('LietotajsData', 'GrupasData',
'AudzeknisData', 'VecakiData', 'TerpiData', 'KoncertiData', 'KoncertiNumuri',
'nodarbibuSkaitis', 'grupuSkaitis'));
}

```

PedagogiNumursController.php

```
<?php

/**
 * PedagogiNumursController nodrošina funkcijas pedagogu pievienošanai un
 * dzēšanai no numuriem.
 * Tas ietver metodes pedagogu pievienošanai numuriem, pedagogu pievienošanas
 * skatu ģenerēšanai un pedagogu dzēšanai no numuriem.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: PedagogNumurs_tabula, Numurs_tabula, Lietotajs_tabula,
 * Lomas_tabula
 * Izmantotie skati: pedagogiNumurs, pievPedPedagogiNumurs
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\PedagogNumurs_tabula;
use App\Models\Numurs_tabula;
use App\Models\Lietotajs_tabula;
use App\Models\Lomas_tabula;

class PedagogiNumursController extends Controller
{
    /**
     * Ģenerē skatu ar pedagogu sarakstu, kas ir saistīti ar konkrētu numuru.
     */
    public function PedagogiNumursIndex($NumursID)
    {
        $LietotajsData = Lietotajs_tabula::where('LomaID', 3)
            ->whereExists(function ($subquery) use ($NumursID) {
                $subquery
                    ->select('*')
                    ->from('pedagognumurs')
                    ->whereRaw('PedagogaNumuraPersonasKods = personasKods')
                    ->where('NumursIDPedagogs', $NumursID);
            })
            ->get();

        $LomasData = Lomas_tabula::all();
        $NumursData = Numurs_tabula::find($NumursID);

        return view('pedagogiNumurs', [
            'LietotajsData' => $LietotajsData,
            'LomasData' => $LomasData,
            'NumursData' => $NumursData,
        ]);
    }

    /**
     * Ģenerē skatu pedagogu pievienošanai numuram.
     */
    public function PievPedPedagogiNumursIndex($NumursID)
    {
        $LietotajsData = Lietotajs_tabula::where('LomaID', 3)
            ->whereNotExists(function ($query) use ($NumursID) {
                $query
                    ->select('*')
                    ->from('pedagognumurs')
            })
            ->get();
    }
}
```

```

        ->whereRaw('PedagogaNumuraPersonasKods = personasKods')
        ->where('NumursIDPedagogs', $NumursID);
    })
    ->get();

    $LomasData = Lomas_tabula::all();
    $NumursData = Numurs_tabula::find($NumursID);

    return view('pievPedPedagogiNumurs', [
        'NumursData' => $NumursData,
        'LietotajsData' => $LietotajsData,
        'LomasData' => $LomasData,
    ]);
}

/**
 * Pievieno pedagogu numuram.
 */
public function PedagogiNumursDataInsert($NumursID, $personasKods)
{
    $isPedagogiNumursInsertSuccess = PedagogNumurs_tabula::insert([
        'PedagogaNumuraPersonasKods' => $personasKods,
        'NumursIDPedagogs' => $NumursID,
    ]);

    if ($isPedagogiNumursInsertSuccess) {
        echo '<script>alert("Pedagogs tika pievienots numuram
;)"</script>';
    } else {
        echo '<script>alert("Pedagogs netika pievienots numuram
:(")</script>';
    }

    return $this->PedagogiNumursIndex($NumursID);
}

/**
 * Dzēš pedagogu no numura.
 */
public function PedagogiNumursDataDelete($NumursID, $personasKods)
{
    $isPedagogiNumursDeleteSuccess = PedagogNumurs_tabula::where([
        ['NumursIDPedagogs', '=', $NumursID],
        ['PedagogaNumuraPersonasKods', '=', $personasKods],
    ])->delete();

    if ($isPedagogiNumursDeleteSuccess) {
        echo '<script>alert("Pedagogs tika izdzēsts no
numura")</script>';
    } else {
        echo '<script>alert("Pedagogs netika izdzēsts no
numura")</script>';
    }

    return $this->PievPedPedagogiNumursIndex($NumursID);
}
}

```


PDFController.php

```
<?php

/**
 * PDFController nodrošina PDF dokumentu ģenerēšanu dažādiem sistēmas datiem,
 * izmantojot DomPDF bibliotēku.
 * Tas ietver metodes grupu, tērpu un koncerta datu eksportēšanai uz PDF
 * formātu.
 *
 * Nosaukums: App\Http\Controllers
 * Izmantotie modeļi: Koncerts_tabula, KoncertsNumurs_tabula, Numurs_tabula,
 * AudzeknisNumurs_tabula, PedagogNumurs_tabula, Lietotajs_tabula,
 * Grupas_tabula, Terpi_tabula
 * Izmantotie skati: pdf.grupas_saraksts, pdf.terpi_saraksts,
 * pdf.koncerts_saraksts, pdf.koncerts_kostimi_saraksts, pdf.pedagogs_terpi,
 * pdf.pedagogs_numuri, pdf.audzeknis_kostimi_saraksts
 */

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Barryvdh\DomPDF\Facade\Pdf;
use App\Models\Koncerts_tabula;
use App\Models\KoncertsNumurs_tabula;
use App\Models\Numurs_tabula;
use App\Models\AudzeknisNumurs_tabula;
use App\Models\PedagogNumurs_tabula;
use App\Models\Lietotajs_tabula;
use App\Models\Grupas_tabula;
use App\Models\Terpi_tabula;

class PDFController extends Controller
{
    /**
     * Ģenerē PDF failu ar grupas sarakstu.
     */
    public function generateGrupasPDF($GrupasNosaukums)
    {
        $grupa = Grupas_tabula::where('GrupasNosaukums', $GrupasNosaukums)-
        >first();

        $audzekni = Lietotajs_tabula::whereIn('LomaID', [1])
        ->whereExists(function ($query) use ($GrupasNosaukums) {
            $query->select('*')
            ->from('audzeknisgrupa')
            ->whereRaw('AudzeknaGrupasPersonasKods = personasKods')
            ->where('GrupasAudzeknaNosaukums', $GrupasNosaukums);
        })
        ->with('vecaki')
        ->get();

        $pedagogs = Lietotajs_tabula::whereIn('LomaID', [3])
        ->whereExists(function ($query) use ($GrupasNosaukums) {
            $query->select('*')
            ->from('pedagogsgrupa')
            ->whereRaw('PedagogaPersonasKods = personasKods')
            ->where('GrupasPedagogaNosaukums', $GrupasNosaukums);
        })
        ->first();
    }
}
```

```

    $data = [
        'grupa' => $grupa,
        'audzekni' => $audzekni,
        'pedagogs' => $pedagogs
    ];

    $pdf = PDF::loadView('pdf.grupas_saraksts', $data);
    return $pdf->download('grupas_saraksts_'. $GrupasNosaukums.'.pdf');
}

/**
 * Ģenerē PDF failu ar visu tērpju sarakstu.
 */
public function generateTerpiPDF()
{
    $terpi = Terpi_tabula::all();
    $terpiCount = $terpi->count();

    $data = [
        'terpi' => $terpi,
        'terpiCount' => $terpiCount
    ];

    $pdf = PDF::loadView('pdf.terpi_saraksts', $data);
    return $pdf->download('terpi_saraksts.pdf');
}

/**
 * Ģenerē PDF failu ar konkrēta koncerta sarakstu.
 */
public function generateKoncertsPDF($KoncertsID)
{
    $koncerts = Koncerts_tabula::find($KoncertsID);
    $numuri = KoncertsNumurs_tabula::where('KoncertsIDKoncNumurs',
$KoncertsID)
        ->orderBy('KartasNumurs')
        ->get();

    foreach ($numuri as $numurs) {
        $numurs->audzekni =
AudzeknisNumurs_tabula::where('NumursIDAudzeknis', $numurs-
>NumursIDKoncNumurs)
            ->join('lietotajs',
'audzeknisnumurs.AudzeknaNumuraPersonasKods', '=', 'lietotajs.personasKods')
            ->leftJoin('audzeknisgrupa',
'audzeknisnumurs.AudzeknaNumuraPersonasKods', '=',
'audzeknisgrupa.AudzeknaGrupasPersonasKods')
            ->leftJoin('grupa', 'audzeknisgrupa.GrupasAudzeknaNosaukums',
'=', 'grupa.GrupasNosaukums')
            ->select('lietotajs.Vards', 'lietotajs.Uzvards',
'grupa.GrupasNosaukums')
            ->get();

        $numurs->pedagogi =
PedagogsNumurs_tabula::where('NumursIDPedagogs', $numurs->NumursIDKoncNumurs)
            ->join('lietotajs',
'pedagogsnumurs.PedagogaNumuraPersonasKods', '=', 'lietotajs.personasKods')
            ->select('lietotajs.Vards', 'lietotajs.Uzvards')
            ->get();
    }
}

```

```

    $data = [
        'koncerts' => $koncerts,
        'numuri' => $numuri
    ];

    $pdf = PDF::loadView('pdf.koncerts_saraksts', $data);
    return $pdf->download('koncerts_saraksts_'. $koncerts-
>Nosaukums.'.pdf');
}

/**
 * Eksportē PDF failu ar konkrēta koncerta kostīmu sarakstu.
 */
public function exportKoncertsKostimiPDF($koncertsID)
{
    $koncerts = Koncerts_tabula::findOrFail($koncertsID);
    $numuri = KoncertsNumurs_tabula::where('KoncertsIDKoncNumurs',
$koncertsID)->get();

    foreach ($numuri as $numurs) {
        $numurs->numurs = Numurs_tabula::with('kostims')->find($numurs-
>NumursIDKoncNumurs);
    }

    $data = [
        'koncerts' => $koncerts,
        'numuri' => $numuri
    ];

    $pdf = PDF::loadView('pdf.koncerts_kostimi_saraksts', $data);
    return $pdf->download('koncerts_kostimi_saraksts_'. $koncerts-
>Nosaukums.'.pdf');
}

/**
 * Eksportē PDF failu ar pedagoga tērpu sarakstu konkrētam koncertam.
 */
public function exportPedagogsTerpiPDF($personasKods, $koncertsID)
{
    $pedagogs = Lietotajs_tabula::where('personasKods', $personasKods)-
>first();
    $koncerts = Koncerts_tabula::findOrFail($koncertsID);

    $numuri = KoncertsNumurs_tabula::where('KoncertsIDKoncNumurs',
$koncertsID)
        ->whereHas('numurs.pedagogi', function($query) use
($personasKods) {
            $query->where('PedagogaNumuraPersonasKods', $personasKods);
        })
        ->with('numurs.kostims')
        ->get();

    $data = [
        'pedagogs' => $pedagogs,
        'koncerts' => $koncerts,
        'numuri' => $numuri
    ];

    $pdf = PDF::loadView('pdf.pedagogs_terpi', $data);
    return $pdf->download('pedagogs_terpi_'. $pedagogs-
>Vards.'. $pedagogs->Uzvards.'.pdf');
}

```

```

    }

    /**
     * Eksportē PDF failu ar pedagoga numuru sarakstu konkrētam koncertam.
     */
    public function exportPedagogsNumuriPDF($personasKods, $koncertsID)
    {
        $pedagogs = Lietotajs_tabula::where('personasKods', $personasKods)-
>first();
        $koncerts = Koncerts_tabula::findOrFail($koncertsID);

        $numuri = KoncertsNumurs_tabula::where('KoncertsIDKoncNumurs',
$koncertsID)
        ->whereHas('numurs.pedagogi', function($query) use
($personasKods) {
            $query->where('PedagogaNumuraPersonasKods', $personasKods);
        })
        ->orderBy('KartasNumurs')
        ->with(['numurs', 'numurs.audzekni'])
        ->get();

        $data = [
            'pedagogs' => $pedagogs,
            'koncerts' => $koncerts,
            'numuri' => $numuri
        ];

        $pdf = PDF::loadView('pdf.pedagogs_numuri', $data);
        return $pdf->download('pedagogs_numuri_'.$pedagogs-
>Vards.''.$pedagogs->Uzvards.'.pdf');
    }

    /**
     * Eksportē PDF failu ar audzēkņa tērpu sarakstu konkrētam koncertam.
     */
    public function exportAudzeknisTerpiPDF($personasKods, $koncertsID)
    {
        $audzeknis = Lietotajs_tabula::where('personasKods', $personasKods)-
>first();
        $koncerts = Koncerts_tabula::findOrFail($koncertsID);
        $numuri = KoncertsNumurs_tabula::where('KoncertsIDKoncNumurs',
$koncertsID)
        ->whereHas('numurs.audzekni', function($query) use
($personasKods) {
            $query->where('AudzeknaNumuraPersonasKods', $personasKods);
        })
        ->get();

        foreach ($numuri as $numurs) {
            $numurs->numurs = Numurs_tabula::with('kostims')->find($numurs-
>NumursIDKoncNumurs);
        }

        $data = [
            'audzeknis' => $audzeknis,
            'koncerts' => $koncerts,
            'numuri' => $numuri
        ];

        $pdf = PDF::loadView('pdf.audzeknis_kostimi_saraksts', $data);
    }

```

```
        return $pdf->download('audzeknis_kostimi_saraksts_'. $audzeknis-  
>Vards.''. $audzeknis->Uzvards.'.pdf');  
    }  
}
```

ViewTerpiController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Terpi_tabula;

/**
 * ViewTerpiController nodrošina funkcijas tērpu datu skatīšanai, rediģēšanai
 un meklēšanai.
 */
* Nosaukums: App\Http\Controllers
* Izmantotie modeļi: Terpi_tabula
* Izmantotie skati: viewTerpi, terpiRediget
*/
class ViewTerpiController extends Controller
{
    /**
     * Ģenerē skatu ar tērpu sarakstu.
     */
    function ViewTerpiIndex()
    {
        $TerpiData = Terpi_tabula::all();
        return view("viewTerpi", ["TerpiData" => $TerpiData]);
    }

    /**
     * Ģenerē skatu tērpa datu rediģēšanai.
     */
    function TerpiRedigetIndex($KostimiID)
    {
        $TerpiData = Terpi_tabula::where("KostimiID", $KostimiID)->get();
        if (count($TerpiData) > 0) {
            return view("terpiRediget", ["td" => $TerpiData[0]]);
        } else {
            return view("terpiRediget", ["td" => null]);
        }
    }

    /**
     * Atjaunina tērpa datus.
     */
    function DataUpdateTerpi(Request $request)
    {
        $KostimiID = $request->KostimiID;
        $Nosaukums = $request->input("Nosaukums");
        $Krasa = $request->input("Krasa");
        $Izmers = $request->input("Izmers");
        $Attels = $request->file("Attels");

        if ($Attels) {
            $imageData = file_get_contents($Attels->getRealPath());
            $base64 = base64_encode($imageData);
        } else {
            $base64 = null;
        }

        $isTerpiUpdateSuccess = Terpi_tabula::where("KostimiID", $KostimiID)-
>update([

```

```

        "Nosaukums" => $request->input("Nosaukums"),
        "Krasa" => $request->input("Krasa"),
        "Izmers" => $request->input("Izmers"),
        "Attels" => $base64,
    ]);

    if ($isTerpiUpdateSuccess) {
        echo '<script>alert("Kostīma dati veiksmīgi rediģēti
;)"</script>';
    } else {
        echo '<script>alert("Kostīma dati netika rediģēti
:(")</script>';
    }

    $TerpiData = Terpi_tabula::all();
    return view("viewTerpi", ["TerpiData" => $TerpiData]);
}

/**
 * Meklē tērpus pēc nosaukuma, krāsas vai izmēra.
 */
function SearchTerpi(Request $request)
{
    $query = $request->input("query");
    $TerpiData = Terpi_tabula::where("Nosaukums", "LIKE", "%" . $query .
"%")
        ->orWhere("Krasa", "LIKE", "%" . $query . "%")
        ->orWhere("Izmers", "LIKE", "%" . $query . "%")
        ->get();

    return view("viewTerpi", ["TerpiData" => $TerpiData]);
}
}

```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

/**
 * AudzekniKostimi_tabula modelis nodrošina audzēkņu un kostīmu attiecību
 * datu pārvaldību.
 *
 * Nosaukums: App\Models
 *
 * Parametri:
 * - $AudzKostID - primārā atslēga
 */
class AudzekniKostimi_tabula extends Model
{
    use HasFactory;

    /**
     * Tabulas nosaukums, kas saistīta ar šo modeli.
     */
    public $table = 'audzeknikostimi';

    /**
     * Primārās atslēgas kolonnas nosaukums.
     */
    public $primaryKey = 'AudzKostID';

    /**
     * Norāda, vai primārā atslēga ir automātiski pieaugoša.
     */
    public $incrementing = false;

    /**
     * Norāda, vai modelim ir atjaunināšanas un izveides laika zīmogi.
     */
    public $timestamps = false;
}
```



```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

/**
 * AudzeknisNumurs_tabula modelis nodrošina audzēkņu un numuru attiecību datu
 pārvaldību.
 *
 * Nosaukums: App\Models
 *
 * Parametri:
 * - $AudzeknisNumursID - primārā atslēga
 */
class AudzeknisNumurs_tabula extends Model
{
    use HasFactory;

    /**
     * Tabulas nosaukums, kas saistīta ar šo modeli.
     */
    public $table = 'audzeknisnumurs';

    /**
     * Primārās atslēgas kolonnas nosaukums.
     */
    public $primaryKey = 'AudzeknisNumursID';

    /**
     * Norāda, vai primārā atslēga ir automātiski pieaugoša.
     */
    public $incrementing = false;

    /**
     * Norāda, vai modelim ir atjaunināšanas un izveides laika zīmogi.
     */
    public $timestamps = false;
}
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

/**
 * BernsVecaks_tabula modelis nodrošina bērnu un vecāku attiecību datu
 pārvaldību.
 *
 * Nosaukums: App\Models
 *
 * Parametri:
 * - $BernsVecaksID - primārā atslēga
 */
class BernsVecaks_tabula extends Model
{
    use HasFactory;

    /**
     * Tabulas nosaukums, kas saistīta ar šo modeli.
     */
    public $table = 'bernsvecaks';

    /**
     * Primārās atslēgas kolonnas nosaukums.
     */
    public $primaryKey = 'BernsVecaksID';

    /**
     * Norāda, vai primārā atslēga ir automātiski pieaugoša.
     */
    public $incrementing = false;

    /**
     * Norāda, vai modelim ir atjaunināšanas un izveides laika zīmogi.
     */
    public $timestamps = false;
}
```

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

/**
 * Lietotajs_tabula modelis nodrošina lietotāja datu pārvaldību.
 *
 * Nosaukums: App\Models
 *
 * Parametri:
 * - $personasKods - primārā atslēga
 */
class Lietotajs_tabula extends Model
{
    use HasFactory;

    /**
     * Tabulas nosaukums, kas saistīta ar šo modeli.
     */
    public $table = 'lietotajs';

    /**
     * Primārās atslēgas kolonnas nosaukums.
     */
    public $primaryKey = 'personasKods';

    /**
     * Norāda, vai primārā atslēga ir automātiski pieaugoša.
     */
    public $incrementing = false;

    /**
     * Norāda, vai modelim ir atjaunināšanas un izveides laika zīmogi.
     */
    public $timestamps = false;

    /**
     * Attiecība "daudzi pret daudziem" ar Lietotajs_tabula modeli caur
     * bernsvēcaks tabulu.
     */
    public function vecaki()
    {
        return $this->belongsToMany(Lietotajs_tabula::class, 'bernsvecaks',
        'BernaPersonasKods', 'VecakaPersonasKods');
    }

    /**
     * Attiecība "daudzi pret daudziem" ar Grupas_tabula modeli caur
     * audzeknisgrupa tabulu.
     */
    public function grupa()
    {
        return $this->belongsToMany(Grupas_tabula::class, 'audzeknisgrupa',
        'AudzeknaGrupasPersonasKods', 'GrupasAudzeknaNosaukums');
    }
}

```

Terpi_tabula.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

/**
 * Terpi_tabula modelis nodrošina kostīmu datu pārvaldību.
 *
 * Nosaukums: App\Models
 *
 * Parametri:
 * - $KostimiID - primārā atslēga
 */
class Terpi_tabula extends Model
{
    use HasFactory;

    /**
     * Tabulas nosaukums, kas saistīta ar šo modeli.
     */
    public $table = 'kostimi';

    /**
     * Primārās atslēgas kolonnas nosaukums.
     */
    public $primaryKey = 'KostimiID';

    /**
     * Norāda, vai primārā atslēga ir automātiski pieaugoša.
     */
    public $incrementing = false;

    /**
     * Norāda, vai modelim ir atjaunināšanas un izveides laika zīmogi.
     */
    public $timestamps = false;
}
```

```

<!--Šis HTML dokuments ģenerē PDF failu, kurā ir iekļauts audzēkņa un
koncerta informācija,
kā arī saraksts ar numuriem un to attiecīgajiem kostīmiem.
Dokumentā ir iekļauts logotips, koncerta datums, vieta un ilgums, kā arī
tabula ar numuru detaļām.-->
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Kostīmu Saraksts</title>
  <style>
    body {
      font-family: DejaVu Sans, sans-serif;
    }
    .header {
      text-align: left;
      margin-bottom: 20px;
      display: flex;
      align-items: center;
    }
    .header img {
      max-width: 600px;
      margin-right: 20px;
    }
    table {
      width: 100%;
      border-collapse: collapse;
    }
    th, td {
      border: 1px solid black;
      padding: 8px;
      text-align: left;
    }
  </style>
</head>
<body>
  <div class="header">
    
    </div>
    <h1 style="text-align: center;">{{ $audzeknis->Vards }} {{ $audzeknis-
>Uzvards }} - {{ $koncerts->Nosaukums }}</h1>
    <p style="text-align: center;">{{ $koncerts->Datums }}, {{ $koncerts-
>Vieta }}, {{ $koncerts->Ilgums }}</p>
    <h2>Numuri</h2>
    <table>
      <thead>
        <tr>
          <th>Kārtas numurs</th>
          <th>Nosaukums</th>
          <th>Garums</th>
          <th>Mūzika</th>
          <th>Horeogrāfija</th>
          <th>Kostīms</th>
          <th>Kostīma Attēls</th>
        </tr>
      </thead>
      <tbody>

```

```

@foreach($numuri as $numurs)
    <tr>
        <td>{{ $numurs->KartasNumurs }}</td>
        <td>{{ $numurs->numurs->NumuraNosaukums }}</td>
        <td>{{ $numurs->numurs->Garums }}</td>
        <td>{{ $numurs->numurs->Muzika }}</td>
        <td>{{ $numurs->numurs->Horeografija }}</td>
        <td>{{ $numurs->numurs->kostims->Nosaukums }}</td>
        <td>
            @if($numurs->numurs->kostims->Attels)
                
            @endif
        </td>
    </tr>
@endforeach
</tbody>
</table>
</body>
</html>

```

```

<!--
Šis HTML dokuments nodrošina pedagoga informācijas pārvaldību baletskolas
sistēmā.
Dokuments ietver navigācijas joslu ar saitēm uz dažādām lapām un sekcijas,
kurās tiek attēlota pedagoga informācija, grupas, noliktavas tērpi un
koncerti.
Lietotāji var rediģēt pedagoga datus, pievienot vai dzēst grupas un tērpus,
kā arī eksportēt dažādus sarakstus.
-->
<!DOCTYPE html>
<html lang="lv">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pedagogs</title>
    <link rel="stylesheet" href="{{ asset('css/bootstrap.css') }}">
    <link rel="stylesheet" type="text/css" href="{{ asset('main.css') }}">
</head>
<body>
@if(session('success'))
    @if(session('success') == true)
        <script>alert("Lietotāja dati tika pievienoti ;)");</script>
    @else
        <script>alert("Lietotāja dati netika pievienoti :(");</script>
    @endif
@endif

<!-- Navigācija -->
<nav class="navbar navbar-expand-lg navbar-light">
    <a class="navbar-brand" href="/login">Pedagogs</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
</nav>

<!-- Pedagoga informācija -->
<div class="container">
    <h2>Pedagoga informācija</h2>
    <table class="table">
        <thead>
            <tr>
                <th>Vārds un Uzvārds</th>
                <th>Tālruna numurs</th>
                <th>E-pasts</th>
                <th>Kopējais nodarbību skaits nedēļā</th>
                <th>Kopējais grupu skaits</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>{{ $LietotajsData->Vards }} {{ $LietotajsData->Uzvards
}}</td>
                <td>{{ $LietotajsData->Talrunis }}</td>
                <td>{{ $LietotajsData->Epasts }}</td>
                <td>{{ $nodarbibuSkaitis }}</td>
                <td>{{ $grupuSkaitis }}</td>
            </tr>
        </tbody>
    </table>
</div>

```

```

        </tbody>
    </table>
</div>

<!-- Grupas informācija -->
<div class="container">
    <h2>Grupas</h2>
    <table class="table">
        <thead>
            <tr>
                <th>Nosaukums</th>
                <th>Grafiks</th>
                <th>Nodarbību skaits nedēļā</th>
                <th>Filiale</th>
                <th>Adrese</th>
                <th>Dalībnieki</th>
            </tr>
        </thead>
        <tbody>
            @foreach($GrupasData as $gd)
                <tr>
                    <td>{{ $gd->GrupasNosaukums }}</td>
                    <td>{{ $gd->Grafiks }}</td>
                    <td>{{ $gd->NodSk }}</td>
                    <td>{{ $gd->Filiale }}</td>
                    <td>{{ $gd->Valsts }}, {{ $gd->Pilseta }}, {{ $gd->Rajons
}}, {{ $gd->Iela }}, {{ $gd->Eka }}, {{ $gd->Indekss }}</td>
                    <td>
                        <a href="{{ '/audzekniGrupas'. $gd->GrupasNosaukums }}"
class="btn btn-primary">Saraksts</a>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
</div>

<!-- Pieejamie tērpi -->
<div class="container">
    <h2>Noliktavas tērpi</h2>
    <a href="{{ route('exportTerpiPDF') }}" class="btn btn-primary">Eksportēt
sarakstu</a>
    <table class="table">
        <thead>
            <tr>
                <th>Nosaukums</th>
                <th>Izmērs</th>
                <th>Krāsa</th>
                <th>Attēls</th>
                <th>Izdalīšana</th>
                <th>Savākšana</th>
            </tr>
        </thead>
        <tbody>
            @foreach($TerpiData as $terps)
                <tr>
                    <td>{{ $terps->Nosaukums }}</td>
                    <td>{{ $terps->Izmers }}</td>
                    <td>{{ $terps->Krasa }}</td>
                    <td>

```



```


</td>
<td><a href="{{ url('/iedalitTerpi'.$terps->KostimiID)
}}" class="btn btn-primary">Izdalīt</a></td>
<td><a class="btn btn-primary" href="{{
url('/savaktTerpi'.$terps->KostimiID) }}">Savākt</a></td>
</tr>
@endforeach
</tbody>
</table>
</div>

<!-- Koncerti informācija -->
<div class="container">
<h2>Koncerti</h2>
@foreach($KoncertiData as $koncerts)
<h4>{{ $koncerts->Nosaukums }}</h4>
<p>{{ $koncerts->Datums }}, {{ $koncerts->Vieta }}, {{ $koncerts-
>Ilgums }}</p>
<a href="{{ url('/exportPedagogsNumuriPDF/' . session('personasKods')
. '/' . $koncerts->KoncertsID) }}" class="btn btn-primary mb-3">Eksportēt
numuru sarakstu</a>
<a href="{{ url('/exportPedagogsTerpiPDF/' . session('personasKods')
. '/' . $koncerts->KoncertsID) }}" class="btn btn-primary mb-3">Eksportēt
tērpu sarakstu</a>
<h5>Mani koncerta numuri:</h5>
<table class="table">
<thead>
<tr>
<th>Kārtas numurs</th>
<th>Nosaukums</th>
<th>Izpildītāji</th>
<th>Garums</th>
<th>Mūzika</th>
<th>Horeogrāfija</th>
<th>Pedagogi</th>
<th>Kostīmi</th>
</tr>
</thead>
<tbody>
@foreach($KoncertiNumuri as $numurs)
@if($numurs->KoncertsIDKoncNumurs == $koncerts-
>KoncertsID)
<tr>
<td>{{ $numurs->KartasNumurs }}</td>
<td>{{ $numurs->numurs->NumuraNosaukums }}</td>
<td>
@foreach($numurs->numurs->audzekni as
$audzeknis)
{{ $audzeknis->Vards }} {{ $audzeknis-
>Uzvards }}<br>
@endforeach
</td>
<td>{{ $numurs->numurs->Garums }}</td>
<td>{{ $numurs->numurs->Muzika }}</td>
<td>{{ $numurs->numurs->Horeografija }}</td>
<td>
@foreach($numurs->numurs->pedagogi as
$pedagogs)

```

```

                                {{ $pedagogs->Vards }} {{ $pedagogs->
>Uzvards }}<br>
                                @endforeach
                                </td>
                                <td>{{ $numurs->numurs->kostims->Nosaukums
}}</td>
                                </tr>
                                @endif
                                @endforeach
                                </tbody>
                                </table>
                                @endforeach
                                </div>

<div class="container">
    <h2>Lietotāja datu atjaunošana</h2>
    @if (isset($LietotajsData) && $LietotajsData !== null)
        <form action="{{ url('dataUpdateLietotaji/'.$LietotajsData->
personasKods) }}" method="post" enctype="multipart/form-data">
            @csrf
            <div class="form-group">
                <label for="Vards">Vārds:</label>
                <input type="text" class="form-control" name="Vards" value="{{
$LietotajsData->Vards }}">
            </div>
            <div class="form-group">
                <label for="Uzvards">Uzvārds:</label>
                <input type="text" class="form-control" name="Uzvards"
value="{{ $LietotajsData->Uzvards }}">
            </div>
            <div class="form-group">
                <label for="Epasts">E-pasts:</label>
                <input type="email" class="form-control" name="Epasts"
value="{{ $LietotajsData->Epasts }}">
            </div>
            <div class="form-group">
                <label for="Parole">Parole:</label>
                <input type="password" class="form-control" name="Parole"
value="{{ $LietotajsData->Parole }}">
            </div>
            <div class="form-group">
                <label for="Talrunis">Tālrunis:</label>
                <input type="tel" class="form-control" name="Talrunis"
value="{{ $LietotajsData->Talrunis }}">
            </div>
            <button type="submit" class="btn btn-primary">Saglabāt</button>
        </form>
    @endif
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpu"></script>
</body>
</html>

```

```

<!--
Šis HTML dokuments nodrošina numuru saraksta pārvaldības funkcionalitāti
baletskolas sistēmā.
Dokuments ietver navigācijas joslas ar saitēm uz dažādām lapām un tabulu,
kurā ir redzami lietotāji, to lomas, grupas un iespēja dzēst audzēkņus no
numuriem.
Navigācijas joslas un darbības ir atkarīgas no lietotāja lomas, lai
nodrošinātu atbilstošu piekļuvi un funkcionalitāti.
-->
<!DOCTYPE html>
<html lang="lv">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Numuru saraksts</title>
  <link rel="stylesheet" href="{{ asset('css/bootstrap.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ asset('main.css') }}">
</head>
<body>
  <!--Navigācija-->
  <nav class="navbar navbar-expand-lg navbar-light">
    <a class="navbar-brand" href="/addNumurs">Numuri</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="/addTerpi">tērpi</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/">lietotāji</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/addGrupas">grupas</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/addKoncerts">koncerti</a>
        </li>
      </ul>
    </div>
  </nav>
  <nav class="navbar navbar-expand-lg navbar-light navbar-lighter">
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
          <a class="nav-link" href="/addNumurs">Pievienot
numuru</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" style="color:#3a88fe;"
href="/viewNumurs">Numuru saraksts</a>
        </li>

```

```

        </ul>
      </div>
    </nav>
    <nav class="navbar navbar-expand-lg navbar-light navbar-brighter">
      <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ml-auto">
          <a class="navbar-brand" style="color:#3a88fe;" href="{ {
'/audzekniNumurs'. $NumursData->NumursID } }">{{ $NumursData-
>NumuraNosaukums}}</a>
          <li class="nav-item">
            <a class="nav-link" href="{ {
'/pievAudzAudzekniNumurs'. $NumursData->NumursID } }">Pievienot audzēkni</a>
          </li>
        </ul>
      </div>
    </nav>
    <div class="container">
      <table class="table">
        <thead>
          <tr>
            <th>Loma</th>
            <th>Vārds un uzvārds</th>
            <th>Grupa</th>
            <th>Dzēšana</th>
          </tr>
        </thead>
        <tbody>
          @foreach($LietotajsData as $ld)
            <tr>
              <td>
                @foreach($LomasData as $lomd)
                  @if($lomd->LomaID == $ld->LomaID)
                    {{ $lomd->LomasNosaukums}}
                  @endif
                @endforeach
              </td>
              <td><a href="{ { url('/audzeknis'. $ld->personasKods)
}}">{{ $ld->Vards}} {{ $ld->Uzvards}}</a></td>
              <td>{{ $ld->grupa}}</td>
              <td>
                @if($ld->LomaID == 1)
                  <a class="btn btn-danger"
href="/dataDeleteAudzekniNumurs{{ $NumursData->NumursID}}/{{ $ld-
>personasKods}}">Dzēst no numura</a>
                @endif
              </td>
            </tr>
          @endforeach
        </tbody>
      </table>
    </div>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpu"></script>
  </body>
</html>

```

```

<!--
Šis HTML dokuments nodrošina informācijas par audzēkni pārvaldību baletskolas
sistēmā.
Dokuments ietver navigācijas joslu ar saitēm uz dažādām lapām un dažādām
sadaļām, kas attēlo informāciju par audzēkni, grupām, pedagogiem, vecākiem,
kostīmiem un koncertiem.
Navigācijas joslas un darbības ir atkarīgas no lietotāja lomas, lai
nodrošinātu atbilstošu piekļuvi un funkcionalitāti.
-->
<!DOCTYPE html>
<html lang="lv">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dzēst grupu</title>
    <link rel="stylesheet" href="{{ asset('css/bootstrap.css') }}">
    <link rel="stylesheet" type="text/css" href="{{ asset('main.css') }}">
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-light">
      <a class="navbar-brand" href="/addGrupas">Grupas</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <a class="nav-link" href="/addTerpi">tērpi</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/">lietotāji</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/addNumurs">numuri</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/addKoncerts">koncerti</a>
          </li>
        </ul>
      </div>
    </nav>
    <nav class="navbar navbar-expand-lg navbar-light navbar-lighter">
      <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item active">
            <a class="nav-link" href="/addGrupas">pievienot grupu</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" style="color:#3a88fe;"
href="/viewGrupas">grupu saraksts</a>
          </li>
        </ul>
      </div>

```

```

</nav>
<div class="container">
  <div class="row pt-5">
    <div class="col-md-4"></div>
    <div class="col-md-4">
      <div class="card">
        <h4 class="card-header">Izdzēst šo grupu?</h4>
        <div class="card-body">
          <a href="{{ url('/viewGrupas')}}" class="btn btn-sm btn-
primary">Nē</a>
          <a href="{{ url('deleteGrupasData' . $GrupasNosaukums)}}"
class="btn btn-sm btn-danger">Jā</a>
        </div>
      </div>
    </div>
    <div class="col-md-4"></div>
  </div>
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpu">

```

```

<!--
Šis HTML dokuments nodrošina formu tērpa pievienošanai baletskolas sistēmā.
Dokuments ietver navigācijas joslu ar saitēm uz dažādām lapām un formu, kurā
var ievadīt tērpa nosaukumu, krāsu, izmēru un attēlu.
Forma izmanto CSRF aizsardzību, lai nodrošinātu drošību pret krāpnieciskām
darbībām.
-->

<!DOCTYPE html>
<html lang="lv">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tērpu pievienošana</title>
  <link rel="stylesheet" href="{{ asset('css/bootstrap.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ asset('main.css') }}">
</head>
<body>
  <!--Navigācija-->
  <nav class="navbar navbar-expand-lg navbar-light">
    <a class="navbar-brand" href="/addTerpi">Tērpi</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="/">Lietotāji</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/addGrupas">Grupas</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/addNumurs">Numuri</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/addKoncerts">Koncerti</a>
        </li>
      </ul>
    </div>
  </nav>
  <nav class="navbar navbar-expand-lg navbar-light navbar-lighter">
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
          <a class="nav-link" style="color:#3a88fe;"
href="/addTerpi">Pievienot tērpu</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/viewTerpi">Tērpu saraksts</a>
        </li>
      </ul>
    </div>
  </nav>

```

```

</nav>
<!--Forma-->
<div class="container">
  <form action="dataInsertTerpi" method="post" enctype="multipart/form-
data"> @csrf
    <div class="form-group">
      <label for="Nosaukums">Nosaukums:</label>
      <input type="text" class="form-control" name="Nosaukums"
placeholder="piem. svārki" required autofocus>
    </div>
    <div class="form-group">
      <label for="Krasa">Krāsa:</label>
      <input type="text" class="form-control" name="Krasa"
placeholder="piem. rozā" required autofocus>
    </div>
    <div class="form-group">
      <label for="Izmers">Izmērs:</label>
      <input type="text" class="form-control" name="Izmers"
placeholder="piem. 165 cm" required autofocus>
    </div>
    <div class="form-group">
      <label for="Attels">Attēls:</label>
      <input type="file" class="form-control" name="Attels"
placeholder="Pievienojiet attēlu">
    </div>
    <button type="submit" class="btn btn-primary">Pievienot
tērpu</button>
  </form>
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpu"></script>
</body>
</html>

```