

RĪGAS VALSTS TEHNIKUMS

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

PRASĪBU DOKUMENTS

“Brīvā laika pavadīšanas portāls”

Paskaidrojošais raksts lpp.

Audzēknis:

Matīss Šmits DP4-4

Prakses vadītājs:

Igors Litvjakovs

Nodaļas vadītājs:

Normunds Barbāns

Rīga 2024

Anotācija

Kvalifikācijas darbā ir aprakstīts “Brīvā laika pavadīšanas protāls” aktivitāšu pārskatīšanas un tām pieteikšanās process. Sistēma dod iespēju “Brīvā laika pavadīšanas portāls” lietotājiem pārskatīt dažādas aktivitātes, kā arī pieteikties kādām no tām, administrators izveido rezervācijas lai lietotājs var pieteikties no kādām no tām, atkarībā kādā dienā tās notiek. Sistēma tiek izstrādāta ar PHP, HTML, CSS programmēšanas valodām un MYSQL (MyPhpAdmin) relāciju datubāzu vadības sistēmu.

Kvalifikācijas darbs ietver ievadu, uzdevumu nostādni, prasību specifikāciju, uzdevumu risināšanas līdzekļu izvēles pamatojumu, programmatūras produktu modelēšanas un projekta aprakstu, datu struktūru aprakstu, lietotāja ceļvedi, nobeigumu un pielikumus.

Kvalifikācijas darba ievadā ir aprakstīta aktivitāšu aktuālā problēma un tās risinājums. Uzdevuma 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, kādi līdzekļi tiks izmantoti izstrādāšanai un kādiem nolūkiem tie tiek izmantoti. Programmatūras produkta modelēšanas un projektēšanas apraksts sastāv no sistēmas struktūra modeļa, kas ietver sistēmas arhitektūru ER modeli, un funkcionālās sistēmas modeļa, kas satur datu plūsmu modeli. Datu struktūru aprakstā tiek parādīta datu bāzes relāciju shēma, kā arī tabulu struktūra ar aprakstu, kas ietver datu tipu un datu garumu norādīšanu. Lietotāja ceļvedī ir norādītas nepieciešamās sistēmas prasības aparatūrai un programmatūrai, sistēmas instalācija un palaišana, kā arī programmas apraksts, kas paskaidro, kā pareizi lietot sistēmu. Testa piemērā ir dots detalizēt rezervācijas pievienošanas apraksts ar vizuāliem attēliem.

Kvalifikācijas darbs sastāv no -- attēliem, kurā ietilpst -- attēli, -- tabulas un – pielikumi. Pielikumi satur sadaļas “Brīvā laika pavadīšanas portāls” mājaslapas attēlus, funkcionālās dekompozīcijas diagrammu, ER diagrammu, rezervācijas izveidošanas datu plūsmu diagrammu un datu bāzes relāciju shēmu.

Annotation

The Qualification job describes the process of reviewing and logging on to “Briva laika pavadisanas portals” activities. The system enables users of the “Briva laika pavadisanas portals” to review and log on to different activities, and the administrator creates reservations so that the user can log on from one of them depending on the day they take place. The system is developed with PHP, HTML, CSS programming languages, and MYSQL (MyPhpAdmin) relationship database management system.

Qualification work includes introduction, task framework, requirement specification, rationale for selecting task solving features, software product modelling and project description, description of data structures, user guide, end and attachments.

The introduction to the Qualification job describes the current problem of activities and its solution. The Task Framework specifies the tasks that the system will need to perform. The requirement specification shall consist of input and output information as well as functional and non-functional requirements of the system. The rationale for choosing the tools to solve a task indicates which features will be used to develop and for what purposes they are used. The software product modelling and design description consists of a system structure model that includes the system architecture ER model and a functional system model that contains a data flow model. The description of the data structures displays a diagram of database relationships, as well as a table structure with a description that includes specifying the data types and data lengths. The user guide lists the necessary system requirements for hardware and software, system installation and running, and a description of the program that explains how to use the system correctly. The test example gives you a detailed description of adding a reservation with visual images.

The qualification job consists of -- images that include -- images, -- tables, and -- attachments. Attachments contain images of the home page of the “leisure Portal” section, a functional decomposition chart, an ER chart, a booking creation traffic chart, and a database relationship diagram.

SATURS

IEVADS.....	6
1. UZDEVUMA NOSTĀDNE.....	7
2. PRASĪBU SPECIFIKĀCIJA.....	8
2.1. Ieejas un izejas informācijas apraksts.....	8
2.1.1. Ieejas informācijas apraksts	8
2.1.2. Izejas informācijas apraksts	8
2.2. Funkcionālās prasības	9
2.3. Nefunkcionālās prasības	11
3. UZDEVUMA RISINĀŠNAS LĪDZEKĻU IZVĒLES PAMATOJUMS.....	13
4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA	15
4.1. Sistēmas struktūras modelis	15
4.1.1. Sistēmas arhitektūra.....	15
4.1.3. Sistēmas ER modelis.....	16
4.2. Funkcionālais sistēmas modelis.....	18
4.2.1. Datu plūsmu modelis	18
5. DATU STRUKTŪRU APRAKSTS.....	19
6. LIETOTĀJA CEĻVEDIS.....	22
6.1. Sistēmas prasības.....	22
6.2. Sistēmas instalācija un palaišana	23
6.3. Programmas apraksts	26
6.4. Testa piemērs	28
NOBEIGUMS.....	30
INFORMĀCIJAS AVOTI	32
PIELIKUMI	33
Pielikums 1. Pārskatu piemēri ER-diagramma un Datubāzes tabulu saišu shēma:	34
.....	34
Pielikums 2. Datubāzes ekrānattēlu piemēri;.....	35
Pielikums 3. Programmas pirmsteksta kods AdminLapa visas datubāzes saistīšana un programmā visu datubāžu rediģēšana:.....	37

IEVADS

Brīvā laika pavadīšanas portāls ir tiešsaistes platforma, kas izstrādāta, lai lietotājiem sniegtu iespēju atrast un rezervēt dažādas brīvā laika pavadīšanas aktivitātes. Šī portāla galvenais mērķis ir padarīt ērtāku un pieejamāku informāciju par aktivitātēm, kā arī vienkāršot rezervēšanas procesu.

Šis projekts tika izstrādāts kā daļa no kvalifikācijas darba programmēšanas tehniķa specialitātei. Projekta ietvaros tika analizētas lietotāju vajadzības un tirgus prasības, lai izveidotu funkcionālu un lietotājam draudzīgu sistēmu. Portāls nodrošina iespēju lietotājiem reģistrēties, pieteikties un rezervēt aktivitātes, kā arī saņemt atgādinājumus un informāciju par rezervācijām.

Portāla izstrāde tika veikta, izmantojot modernās tīmekļa tehnoloģijas, piemēram, PHP, MySQL, HTML, CSS. Sistēma sastāv no vairākām savstarpēji saistītām komponentēm, kas nodrošina datu apstrādi, lietotāja saskarni un datu glabāšanu.

Projekta laikā tika veikta arī sistēmas testēšana un optimizēšana, lai nodrošinātu tās ātrdarbību un uzticamību. Brīvā laika pavadīšanas portāls ir izstrādāts tā, lai tas būtu viegli pielāgojams un paplašināms nākotnes prasībām un uzlabojumiem.

Šajā dokumentācijā tiks detalizēti aprakstīti portāla izstrādes posmi, tehnoloģiskie risinājumi, izmantotās datu struktūras un lietotāja ceļvedis. Dokumentācija sniegs pilnīgu priekšstatu par izstrādāto sistēmu, tās funkcionalitāti un iespējām.

"Brīvā laika pavadīšanas portāls" izceļas ar daudzveidīgām aktivitātēm un vienkāršu, intuitīvu dizainu, kas padara portālu pieejamu visiem lietotājiem neatkarīgi no viņu tehniskajām prasmēm. Mūsu mērķis ir sniegt jums vislabāko pieredzi un palīdzēt atrast jaunas intereses un hobijus.

Tehniskās prasības

Lai baudītu "Brīvā laika pavadīšanas portālu," jums ir nepieciešams stabils interneta savienojums un atbalstīta pārlūkprogramma (Google Chrome, Mozilla Firefox, utt.).

Sāciet izmantot portālu

Tagad, kad zināt visu nepieciešamo, lai sāktu izmantot "Brīvā laika pavadīšanas portālu," nevilcinieties un iegrimstiet šajā aizraujošajā pasaulē! Reģistrējieties, atklājiet jaunas aktivitātes un izbaudiet savu brīvo laiku ar mums.

1. UZDEVUMA NOSTĀDNE

Kvalifikācijas darba uzdevums ir izveidot “Brīvā laika pavadīšanas portāls” tiešsaistes platformu, kas lietotājiem piedāvā iespēju atrast, rezervēt un pārvaldīt dažādas brīvā laika aktivitātes. Portālam jānodrošina intuitīvs un lietotājam draudzīgs interfeiss, kā arī efektīva datu apstrāde un droša informācijas glabāšana.

Galvenās funkcionalitātes:

- Lietotāju reģistrācija un autentifikācija:
 - Lietotāji var reģistrēties portālā, izveidojot personīgo kontu, izmantojot e-pasta adresi un paroli.
 - Autentificēšanās iespējas, lai nodrošinātu drošu piekļuvi personīgajam kontam.
- Aktivitāšu pārlūkošana un meklēšana:
 - Lietotāji var pārlūkot pieejamās aktivitātes;
 - Detalizēta informācija par katru aktivitāti.
- Rezervāciju veikšana:
 - Lietotāji var rezervēt izvēlētās aktivitātes, norādot vēlamo datumu un laiku;
 - Apstiprinājuma saņemšana par veikto rezervāciju, kā arī iespēja atcelt vai pārplānot rezervāciju.
- Administrēšanas panelis:
 - Administratoriem ir pieejams īpašs interfeiss aktivitāšu pārvaldībai, lietotāju datu apskatei un rezervāciju uzraudzībai;
 - Iespēja pievienot, rediģēt un dzēst aktivitātes, kā arī pārvaldīt lietotāju kontus.
- Paziņojumi un atgādinājumi:
 - Lietotāji saņem paziņojumus par veiksmīgām rezervācijām, izmaiņām aktivitāšu plānos un citiem svarīgiem notikumiem;
 - Iespēja saņemt atgādinājumus par gaidāmajām rezervācijām e-pasta vai īsziņu veidā.
- Atsauksmju un kontaktu:
 - Lietotāji var sniegt atsauksmes un atstāt kontaktos savu ziņu, kas palīdz citiem lietotājiem izvēlēties piemērotākās iespējas.
- Drošība un datu aizsardzība:
 - Nodrošināt lietotāju datu konfidencialitāti un drošību, izmantojot šifrēšanu un citas drošības tehnoloģijas;
 - Regulāri veikt sistēmas uzraudzību un atjaunināšanu, lai novērstu drošības riskus.

2. PRASĪBU SPECIFIKĀCIJA

2.1. Ieejas un izejas informācijas apraksts

2.1.1. Ieejas informācijas apraksts

Lietotāju ievadāmā informācija portālā ietver dažādus datus, kas nepieciešami aktivitāšu rezervēšanai, lietotāju reģistrācijai un autentifikācijai, kā arī aktivitāšu pārvaldībai.

- Lietotāja reģistrācija:
 - Vārds: Lietotāja vārds, līdz 50 simboliem, teksts.
 - Uzvārds: Lietotāja uzvārds, līdz 50 simboliem, teksts.
 - E-pasta adrese: Lietotāja derīga e-pasta adrese, līdz 100 simboliem, teksts.
 - Telefona nummurs: Lietotāja telefona nummurs, līdz 20 simboliem, teksts.
 - Parole: Lietotāja parole, līdz 20 simboliem, teksts.
 - Atkārtotā parole: Lietotāja atkārtotā parole, līdz 20 simboliem, teksts.
- Aktivitāšu pievienošana un rediģēšana (administratoriem):
 - Aktivitātes nosaukums: Līdz 100 simboliem, teksts.
 - Apraksts: Līdz 500 simboliem, teksts.
 - Atrašanās vieta: Līdz 100 simboliem, teksts.
 - Diena un laiks: Diena un laika formāts (dd.mm.yyyy hh).
- Rezervāciju veikšana:
 - Aktivitātes ID: Unikāls identifikators, vesels skaitlis.
 - Lietotāja ID: Unikāls lietotāja identifikators, vesels skaitlis.
 - Rezervācijas diena un laiks: Diena un laika formāts (dd.mm.yyyy hh).

2.1.2. Izejas informācijas apraksts

Rezultātā ģenerētā informācija ietver dažādus datus, kas tiek parādīti lietotājiem vai izmantoti tālākai apstrādei.

- Lietotāja konta informācija:
 - Vārds un uzvārds;
 - E-pasta adrese;
 - Telefona nummurs;
 - Rezervācijas vēsture;
- Aktivitāšu saraksts:
 - Aktivitātes nosaukums;
 - Apraksts;
 - Diena un laiks;
 - Pieejamās vietas;
- Rezervācijas apstiprinājums:
 - Rezervācijas ID;
 - Aktivitātes nosaukums;
 - Dienas un laiks;
 - Rezervācijas statuss (apstiprināta/atcelta);
- Administratīvie pārskati:

- Lietotāju skaits;
- Veikto rezervāciju skaits;
- Populārākās aktivitātes;

2.2. Funkcionālās prasības

Funkcionālās prasības nosaka sistēmas funkcionalitāti, kas ir jānodrošina, lai apmierinātu lietotāju vajadzības. Šeit ir detalizēts prasību saraksts Brīvā laika pavadīšanas portālam:

1. Lietotāju reģistrācija un autentifikācija:

- 1.1. Lietotājiem ir jābūt iespējai reģistrēties, ievadot vārdu, uzvārdu, e-pasta adresi un paroli.
- 1.2. Sistēmai ir jānosūta apstiprinājuma e-pasts pēc reģistrācijas.
- 1.3. Lietotājiem ir jābūt iespējai pieteikties, izmantojot e-pasta adresi un paroli.
- 1.4. Lietotājiem ir jābūt iespējai atiestatīt paroli, ja tā tiek aizmirsta.

2. Lietotāja profils:

- 2.1. Lietotājiem ir jābūt iespējai apskatīt un rediģēt savu profila informāciju.
- 2.2. Lietotājiem ir jābūt iespējai apskatīt savu rezervāciju vēsturi.

3. Aktivitāšu pārlūkošana un meklēšana:

- 3.1. Lietotājiem ir jābūt iespējai pārlūkot pieejamās aktivitātes pēc kategorijām.
- 3.2. Lietotājiem ir jābūt iespējai meklēt aktivitātes pēc nosaukuma, atrašanās vietas un datuma.
- 3.3. Sistēmai ir jāparāda aktivitāšu saraksts ar nosaukumu, aprakstu, atrašanās vietu, datumu un pieejamo vietu skaitu.

4. Aktivitāšu detalizēta apskate:

- 4.1. Lietotājiem ir jābūt iespējai apskatīt detalizētu informāciju par katru aktivitāti, tostarp aprakstu, laiku, vietu un cenu.
- 4.2. Sistēmai ir jāparāda attēli un citi multimediju materiāli, kas saistīti ar aktivitāti.

5. Rezervāciju veikšana:

- 5.1. Lietotājiem ir jābūt iespējai rezervēt izvēlēto aktivitāti, norādot datumu un laiku.
- 5.2. Sistēmai ir jāparāda pieejamo vietu skaits pirms rezervācijas veikšanas.
- 5.3. Pēc veiksmīgas rezervācijas sistēmai ir jāparāda apstiprinājuma ziņojums un jānosūta apstiprinājuma e-pasts.

6. Rezervāciju pārvaldība:

- 6.1. Lietotājiem ir jābūt iespējai apskatīt savas aktīvās rezervācijas.
- 6.2. Lietotājiem ir jābūt iespējai atcelt vai pārplānot savas rezervācijas.

7. Atsauksmju un vērtējumu sistēma:

- 7.1. Lietotājiem ir jābūt iespējai sniegt atsauksmes un vērtējumus par piedalīšanos aktivitātēs.
- 7.2. Sistēmai ir jāparāda vidējais vērtējums pie aktivitātes apraksta.

8. Administrēšanas funkcijas:

- 8.1. Administratoriem ir jābūt iespējai pievienot, rediģēt un dzēst aktivitātes.
- 8.2. Administratoriem ir jābūt iespējai apskatīt lietotāju sarakstu un to rezervācijas.
- 8.3. Administratoriem ir jābūt iespējai ģenerēt pārskatus par populārākajām aktivitātēm un lietotāju aktivitāti.

9. Paziņojumi un atgādinājumi:

- 9.1. Sistēmai ir jānosūta e-pasta paziņojumi par veiksmīgām rezervācijām, izmaiņām un atgādinājumiem.
- 9.2. Lietotājiem ir jābūt iespējai izvēlēties saņemt atgādinājumus par gaidāmajām aktivitātēm.

10. Kontaktu veidlapa:

10.1. Veidlapa: Izveidot kontaktu veidlapu, kurā lietotāji var ievadīt sekojošu informāciju:

- 10.2. Vārds: Teksta lauks (obligāts)
- 10.3. E-pasts: E-pasta lauks (obligāts, ar validāciju, lai nodrošinātu derīgu e-pasta formātu)
- 10.5. Tēma: Teksta lauks (obligāts)
- 10.6. Ziņojums: Teksta laukums (obligāts)

2.3. Nefunkcionālās prasības

1. Veiktspēja:

- 1.1. Sistēmai jābūt spējīgai apstrādāt vismaz 1000 lietotāju pieprasījumus vienlaicīgi.
- 1.2. Atbildes laiks nedrīkst pārsniegt 2 sekundes jebkurai darbībai, izņemot masīvas datu apstrādes operācijas.

2. Drošība:

- 2.1. Lietotāju paroles jāglabā šifrētā veidā, izmantojot spēcīgu šifrēšanas algoritmu.
- 2.2. Jānodrošina datu pārsūtīšana pa SSL/TLS savienojumu, lai aizsargātu datus starp klientu un serveri.
- 2.3. Jāveic regulāri sistēmas drošības auditi, lai identificētu un novērstu iespējamās drošības riskus.

3. Uzturējamība:

- 3.1. Sistēmai jābūt labi dokumentētai, lai to varētu uzturēt un paplašināt citi izstrādātāji.
- 3.2. Kods jāstrukturē modulāri, lai būtu viegli veikt izmaiņas un uzlabojumus.
- 3.3. Jānodrošina, ka sistēmas kļūdas tiek automātiski piefiksētas un ziņotas izstrādātājiem.

4. Lietojamība:

- 4.1. Lietotāja saskarnei jābūt intuitīvai un viegli lietojamai, ievērojot labākās prakses lietojamības dizainā.
- 4.2. Jānodrošina konsekventa navigācija un dizains visā portālā, lai lietotāji varētu viegli atrast nepieciešamo informāciju.
- 4.3. Jāveic lietojamības testēšana ar dažādām lietotāju grupām, lai identificētu un novērstu lietojamības problēmas.

5. Pārnēsamība:

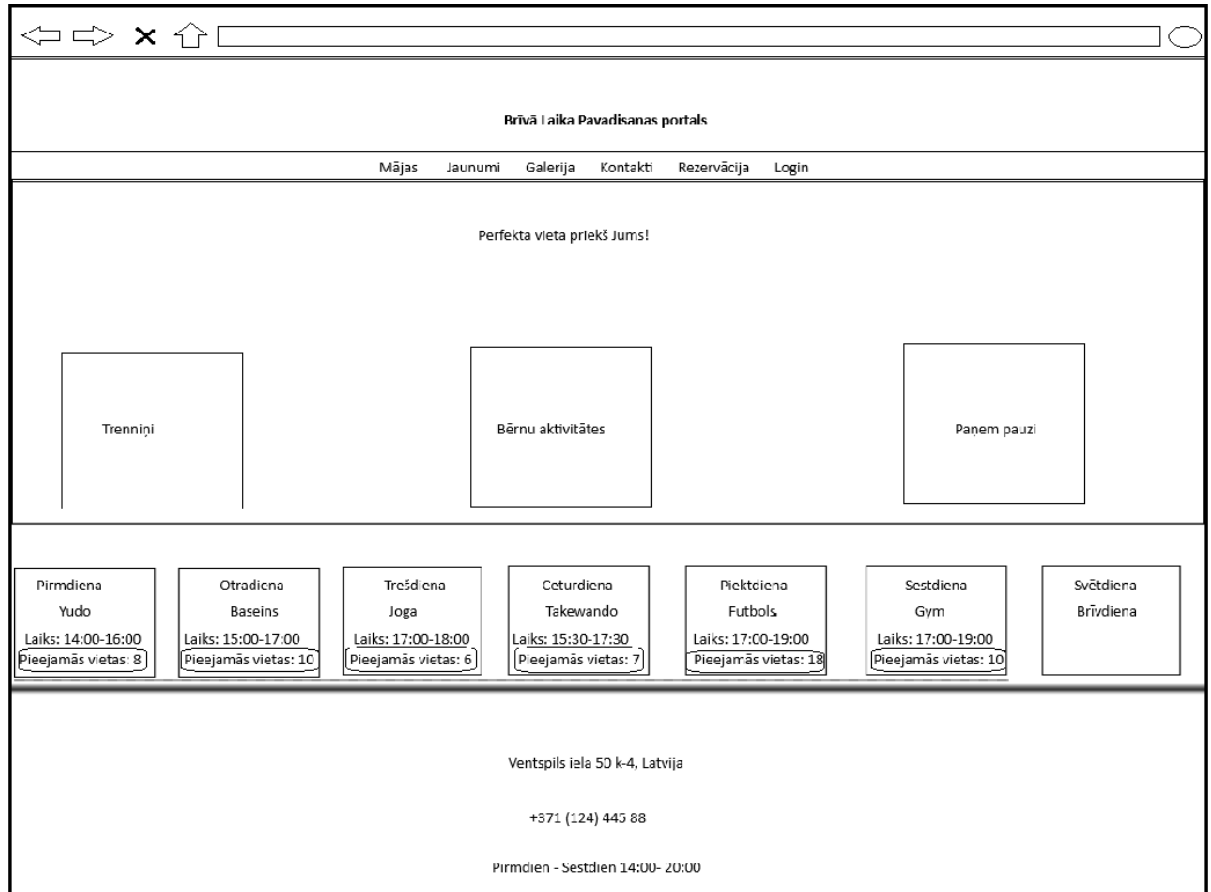
- 5.1 Sistēmai jābūt viegli pārnēsamai uz dažādām platformām un vidiem (piemēram, no izstrādes uz ražošanas vidi).
- 5.2. Jānodrošina, ka sistēmu var darbināt uz dažādām operētājsistēmām (piemēram, Windows, Linux).

6. Pieejamība:

6.1. Sistēmai jābūt pieejamai 24/7, izņemot plānotos apkopju laikus.

6.2. Jānodrošina augsta pieejamība ar rezerves kopēšanas un avārijas atjaunošanas mehānismiem.

6.3. Izvēlnei jābūt izvietotai loga kreisajā pusē (skat. 2.1. att.);



2.1.att. Nefunkcionālo prasību piemērs

7. Skalējamība:

7.1. Sistēmai jābūt viegli skalējamai, lai varētu apstrādāt palielinātu lietotāju un datu daudzumu.

7.2. Jānodrošina horizontāla un vertikāla skalējamība, lai sistēma varētu pielāgoties augošām prasībām.

8. Efektivitāte:

8.1. Sistēmai jābūt optimizētai, lai tā patērētu minimālu resursus, piemēram, atmiņu un procesora jaudu, lai nodrošinātu augstu veiktspēju un efektivitāti.

3. UZDEVUMA RISINĀŠNAS LĪDZEKĻU IZVĒLES PAMATOJUMS

Projekta "Brīvā laika pavadīšanas portāls" izstrādei tika izvēlēti vairāki tehnoloģiskie risinājumi, kas nodrošina augstu veiktspēju, uzticamību un drošību. Šajā sadaļā tiek sniegts pamatojums katra izvēlēta risinājuma izvēlei, balstoties uz to īpašībām un priekšrocībām.

Programmēšanas valodas un ietvari;

PHP:

PHP ir plaši izmantota servera puses programmēšanas valoda, kas ir īpaši piemērota tīmekļa lietotņu izstrādei.

PHP ir viegli apgūstama un piedāvā plašu atbalstu un dokumentāciju, kas nodrošina efektīvu izstrādes procesu.

PHP atbalsta dažādas datu bāzes, tostarp MySQL, kas ir piemērota mūsu projekta vajadzībām.

MySQL (ar phpMyAdmin):

MySQL ir atvērta koda relāciju datu bāzes vadības sistēma, kas piedāvā augstu veiktspēju un uzticamību.

MySQL ir plaši izmantots, nodrošinot lielu atbalstu un resursus izstrādātājiem.

phpMyAdmin ir tīmekļa lietotne, kas paredzēta MySQL datu bāzu pārvaldībai. Tā nodrošina lietotājam draudzīgu interfeisu, kas atvieglo datu bāzu izveidi, pārvaldību un uzturēšanu.

Izstrādes vide un rīki

Apache HTTP Server:

Apache ir populārs atvērta koda tīmekļa serveris, kas nodrošina uzticamu un drošu tīmekļa lietotņu mitināšanu.

Apache piedāvā plašu konfigurācijas iespēju klāstu un labu veiktspēju, kas ir būtiski tīmekļa vietņu darbībai.

JetBrains PhpStorm:

PhpStorm ir spēcīgs IDE, kas piedāvā daudz funkcionalitātes, lai atvieglotu PHP izstrādi, tostarp koda automātisku aizpildīšanu, kļūdu izcelšanu un integrētu atklādošanu.

PhpStorm piedāvā arī labu atbalstu datu bāzes pārvaldībai un integrācijai ar versiju kontroles sistēmām, piemēram, Git.

Git:

Git ir versiju kontroles sistēma, kas nodrošina efektīvu izstrādes procesa pārvaldību, ļaujot izstrādātājiem viegli sekot līdzi izmaiņām kodā un sadarboties komandā.

GitHub ir populāra platforma, kas nodrošina Git repozitoriju mitināšanu, kā arī papildu funkcijas, piemēram, problēmu pārvaldību un nepārtrauktās integrācijas (CI) atbalstu.

Front-end tehnoloģijas;

HTML5:

HTML5 ir jaunākā HTML versija, kas nodrošina uzlabotu funkcionalitāti un labāku atbalstu multivides elementiem, piemēram, video un audio.

HTML5 nodrošina uzlabotu semantiku un labāku veiktspēju, kas ir būtiski tīmekļa lietotņu izstrādē.

CSS3:

CSS3 piedāvā uzlabotas stila lapu iespējas, kas ļauj izstrādātājiem viegli pielāgot un uzlabot tīmekļa vietnes izskatu.

CSS3 nodrošina labāku atbalstu animācijām un pārejām, kas uzlabo lietotāja pieredzi.

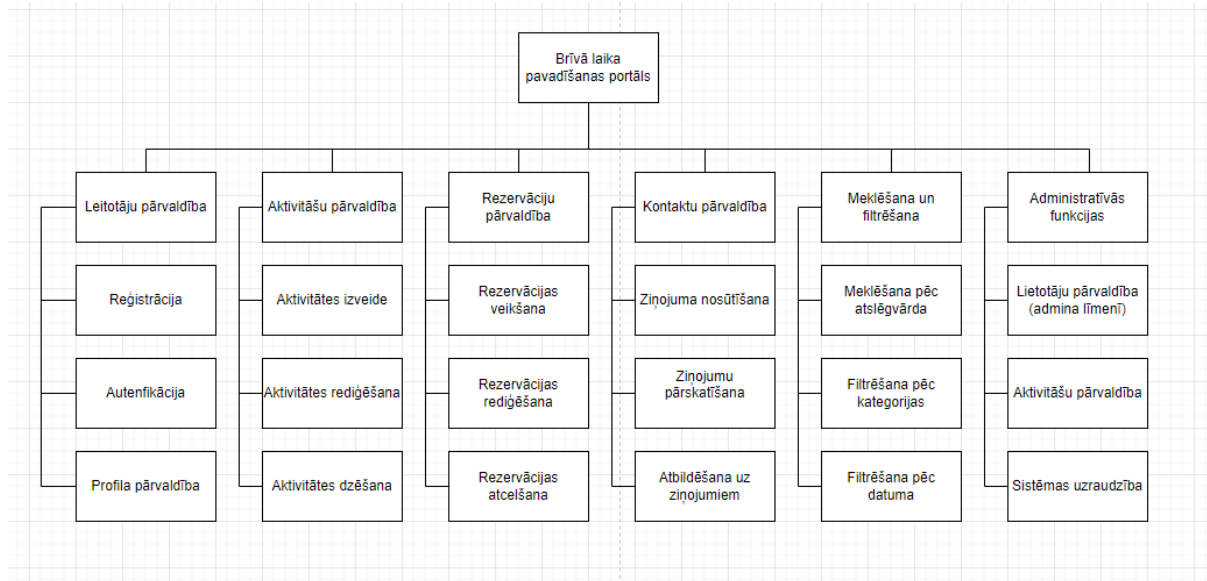
Izvēloties šos risinājumus, mēs nodrošinām, ka "Brīvā laika pavadīšanas portāls" būs efektīvi izstrādāts, viegli uzturams un spēs nodrošināt augstu veiktspēju un drošību. Šie risinājumi kopā veido spēcīgu tehnoloģisko bāzi, kas atbilst mūsdienu tīmekļa lietotņu izstrādes prasībām.

4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

4.1. Sistēmas struktūras modelis

4.1.1. Sistēmas arhitektūra

Brīvā laika pavadīšanas portāla dekompozīcijas diagramma (skat. 4.1. att.) sastāv no 6. apakšsistēmām: lietotāju pārvaldība, aktivitāšu pārvaldība, rezervāciju pārvaldība, kontaktu pārvaldība, meklēšana un filtrēšana, administratīvās funkcijas.



4.1. att. funkcionālā dekompozīcijas diagramma.

Lietotāja saskarne (Front-end):

- HTML, CSS tiek izmantoti, lai izveidotu lietotājam draudzīgu un interaktīvu saskarni.

Lietojumprogrammas loģika (Back-end):

- PHP tiek izmantots, lai apstrādātu lietotāju pieprasījumus, izpildītu biznesa loģiku un mijiedarbotos ar datu bāzi.

Datu bāze:

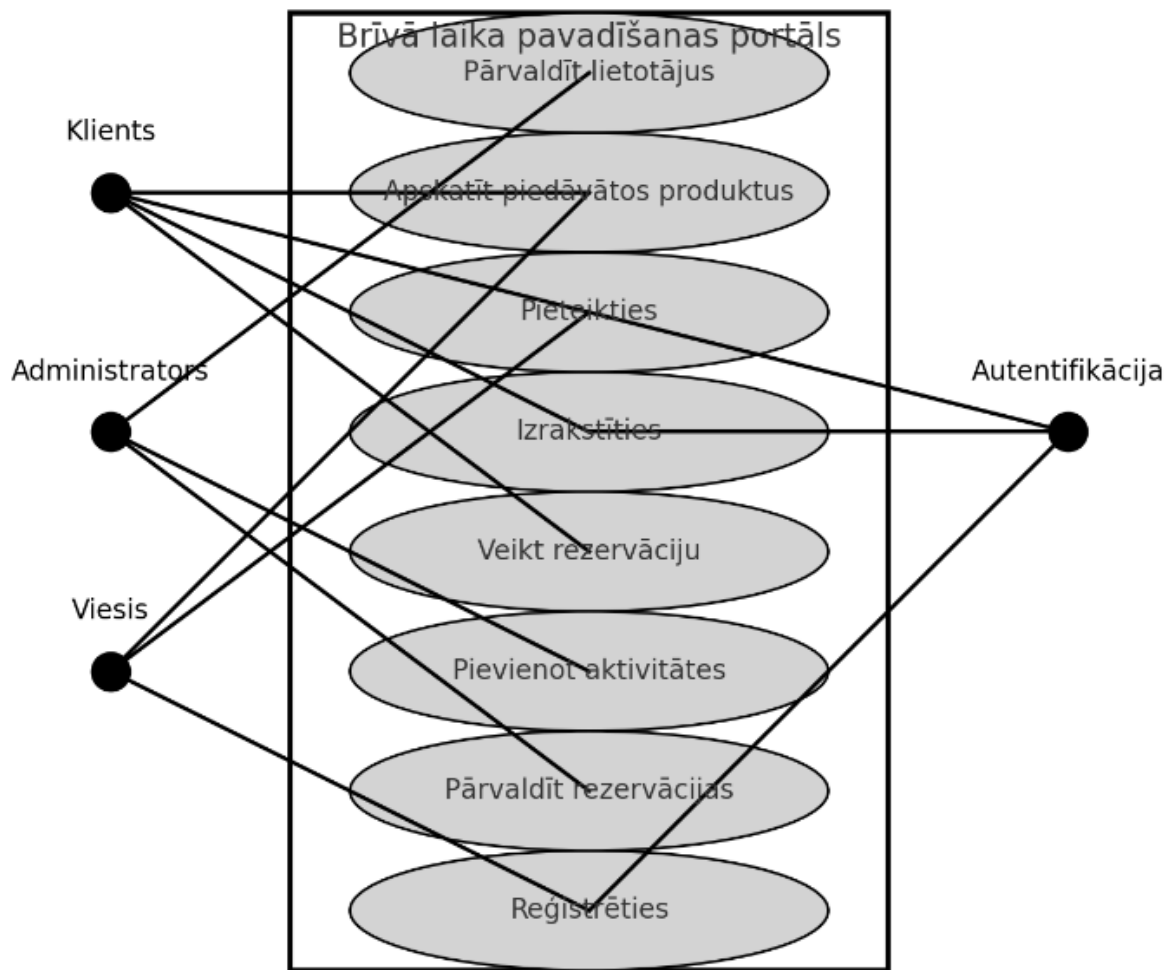
- MySQL ar phpMyAdmin tiek izmantots datu glabāšanai un pārvaldībai.

Serveris:

- Apache HTTP Server nodrošina tīmekļa servera funkcionalitāti, apkalpojot lietotāju pieprasījumus un piegādājot tīmekļa lapas.

4.1.2 Use-Case diagramma

Diagrammu variet aplūkot (skat. 4.1.2. att.), klients var aplūkot piedāvātos produktus(aktivitātes), pieteikties(login), izrakstīties(logout), izveidot kontu(reģistrēties), administratos pārvalda lietotājus, pievieno aktivitātes, pārvalda rezervācijas, viesis, var apskatīt piedāvātos produktus(aktivitātes), pieteikties, reģistrēties.



4.1.2. att. Use-Case diagrammas attēlojums.

4.1.3. Sistēmas ER modelis

ER modelis (Entitāšu-sakaru modelis) attēlo datu bāzes struktūru, galvenās tabulas un to savstarpējās attiecības. Sistēmas ER modelis ietver šādas galvenās entitijas:

Lietotāji: Lauki: lietotāja ID, vārds, uzvārds, e-pasta adrese, parole, telefona nr.;

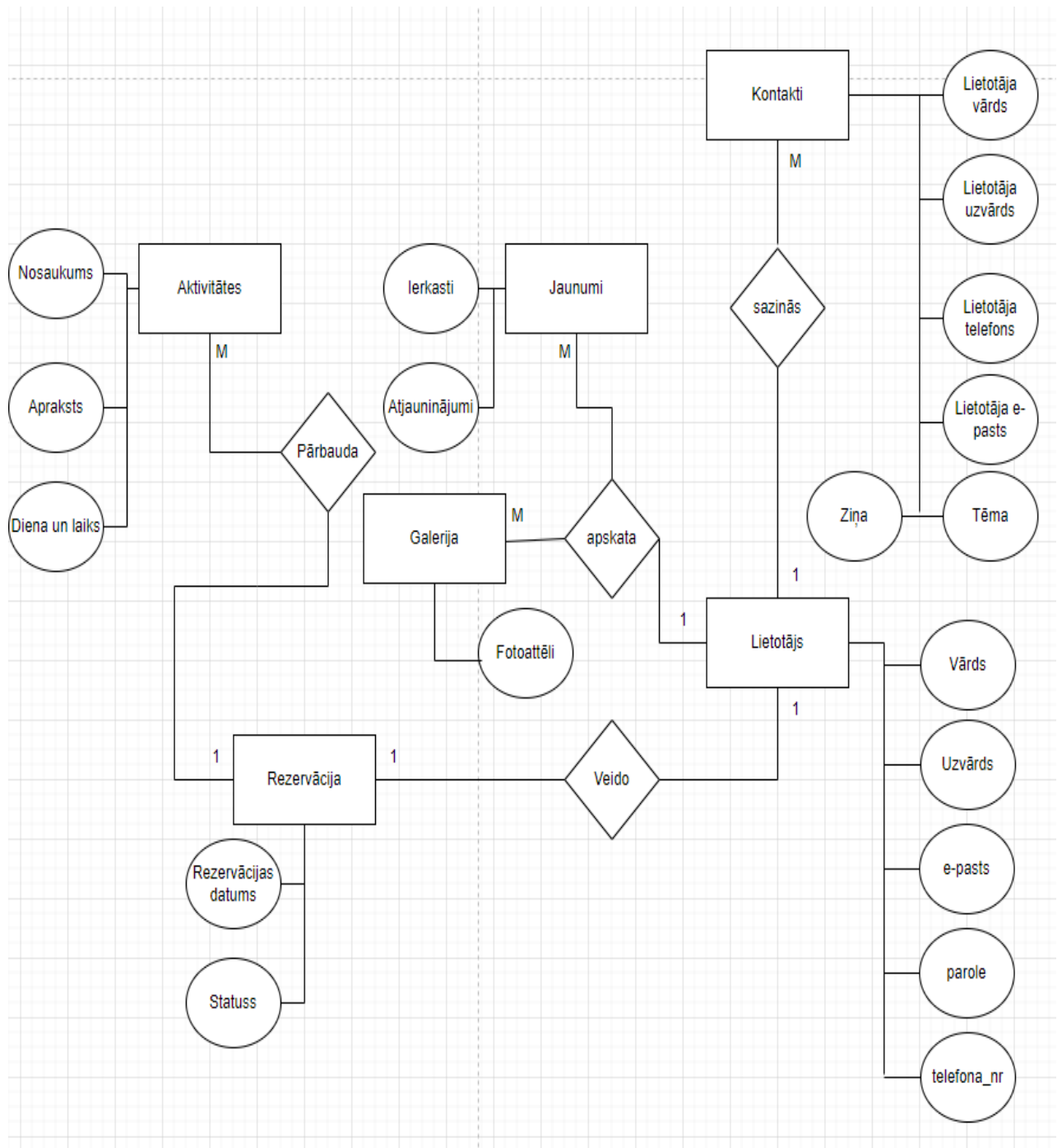
Administratori: Lauki: administratora ID, vārds, uzvārds, e-pasta adrese, parole;

Aktivitātes: Lauki: aktivitātes ID, nosaukums, diena un laiks;

Rezervācijas: Lauki: rezervācijas ID, rezervācijas diena un laiks, statuss;

Kontakti: Lauki: kontaktu ID, lietotāja ID, vārds, uzvārds, e-pasta adrese, ziņojums.

ER diagramma (skat. 4.1.3. att.):



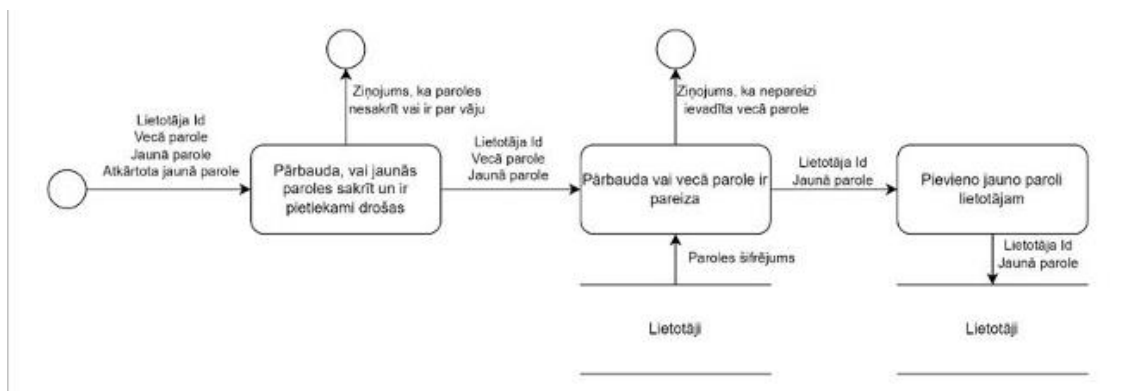
4.2. Funkcionālais sistēmas modelis

4.2.1. Datu plūsmu modelis

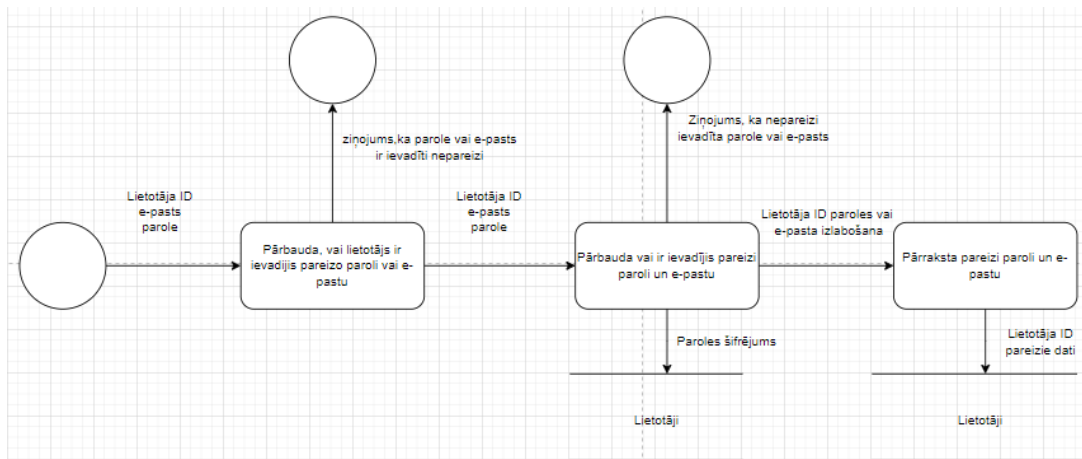
Datu plūsmu modelis attēlo, kā dati pārvietojas caur sistēmu, no ieejas līdz izejai, parādot datu apstrādes procesus un plūsmas. Tas ietver galvenos datu plūsmas posmus no lietotāja pieprasījuma līdz atbildes sniegšanai.

Paroles labošanas modulis (skat. 4.2.1. att.):

Lietotājs ievada veco paroli, jauno paroli un atkārtoto paroli. Tad tiek pārbaudīts, vai jaunā parole abas reizes ievadīta vienādi un vai tā ir pietiekami sarežģīta. Pēc tam tiek pārbaudīta, vai vecā parole ir ievadīta pareizi. Ja kāds no procesiem bija neveiksmīgs, tiek izvadīts kļūdas paziņojums, citādi lietotājam tiek saglabāta jaunā parole.



4.2.1. att. Paroles labošanas moduļa datu plūsmas diagramma.



4.2.2. att.

Ielogošanās moduļa datu plūsmu diagramma.

Ielogošanās modelis (skat. 4.2.2. att.): Lietotājs ievada savu paroli ar e-pastu. Tad tiek pārbaudīts vai parole, vai e-pasts sakrīt ar iepriekš ievadītiem datiem. Pēc tam lietotājs atkārtoti ievada savus datus (paroli un e-pastu). Ja atkal kāds no procesiem bija neveiksmīgs tiek izvadīta kļūda (nepareiza parole vai e-pasts).

5. DATU STRUKTŪRU APRAKSTS

Datubāzes izveidošanai ir nepieciešama datubāzes fiziskā stuktūra. Tā tika iegūta normalizējot ER diagrammu, kuras rezultātā no entitītijām izveidoja relācijas. 5.1. attēlā ir redzama datubāzes tabulu saišu shēma. Kā redzams datubāze sastāv no 5 tabulām, kas satur informāciju par lietotājiem, jaunumiem, kontaktiem, aktivitātēm un rezervāciju.

5.1.1. tabula.

1. Tabulas "Lietotājs" struktūra

Nr.	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	lietotāja_id	int	-	Lietotāja primārā atslēga.
2.	vārds	varchar	255	Lietotāja vārds.
3.	uzvārds	varchar	255	Lietotāja uzvārds.
4.	e-pasts	varchar	255	Lietotāja e-pasta adrese.
5.	parole	varchar	255	Lietotāja parole.
6.	telefona_nr	varchar	15	Lietotāja telefona numurs.

Tabula "Lietotājs" ir saistīta ar tabulu "Kontakti" un "Rezervācija" caur lauku lietotāja_id.

5.1.2. tabula.

2. Tabulas "Administrators" struktūra

Nr.	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	admin_id	int	-	Administratora primārā atslēga.
2.	vārds	varchar	255	Administratora vārds.
3.	uzvārds	varchar	255	Administratora uzvārds.
4.	e-pasts	varchar	255	Administratora e-pasta adrese.
5.	parole	varchar	255	Administratora parole.

Tabula "Administrators" ir saistīta ar tabulu "Aktivitātes" caur lauku admin_id.

5.1.3. tabula.

3. Tabulas "Kontakti" struktūra

Nr.	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	kontakta_id	int	-	Kontakta primārā atslēga.
2.	lietotāja_id	int	-	Saite uz tabulu "Lietotājs".
3.	vārds	varchar	255	Kontakta personas vārds.
4.	uzvārds	varchar	255	Kontakta personas uzvārds.
5.	e-pasts	varchar	255	Kontakta personas e-pasts.
6.	ziņojums	text	-	Kontakta ziņojums.

Tabula "Kontakti" ir saistīta ar tabulu "Lietotājs" caur lauku lietotāja_id.

5.1.4. tabula.

4. Tabulas "Rezervācija" struktūra

Nr.	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	rezervācijas_id	int	-	Rezervācijas primārā atslēga.
2.	lietotāja_id	int	-	Saite uz tabulu "Lietotājs".
3.	aktivitātes_id	int	-	Saite uz tabulu "Aktivitātes".
4.	rezervācijas_datums	datetime	-	Rezervācijas datums.
5.	statuss	varchar	50	Rezervācijas statuss.

Tabula "Rezervācija" ir saistīta ar tabulām "Lietotājs" un "Aktivitātes" caur laukiem lietotāja_id un aktivitātes_id.

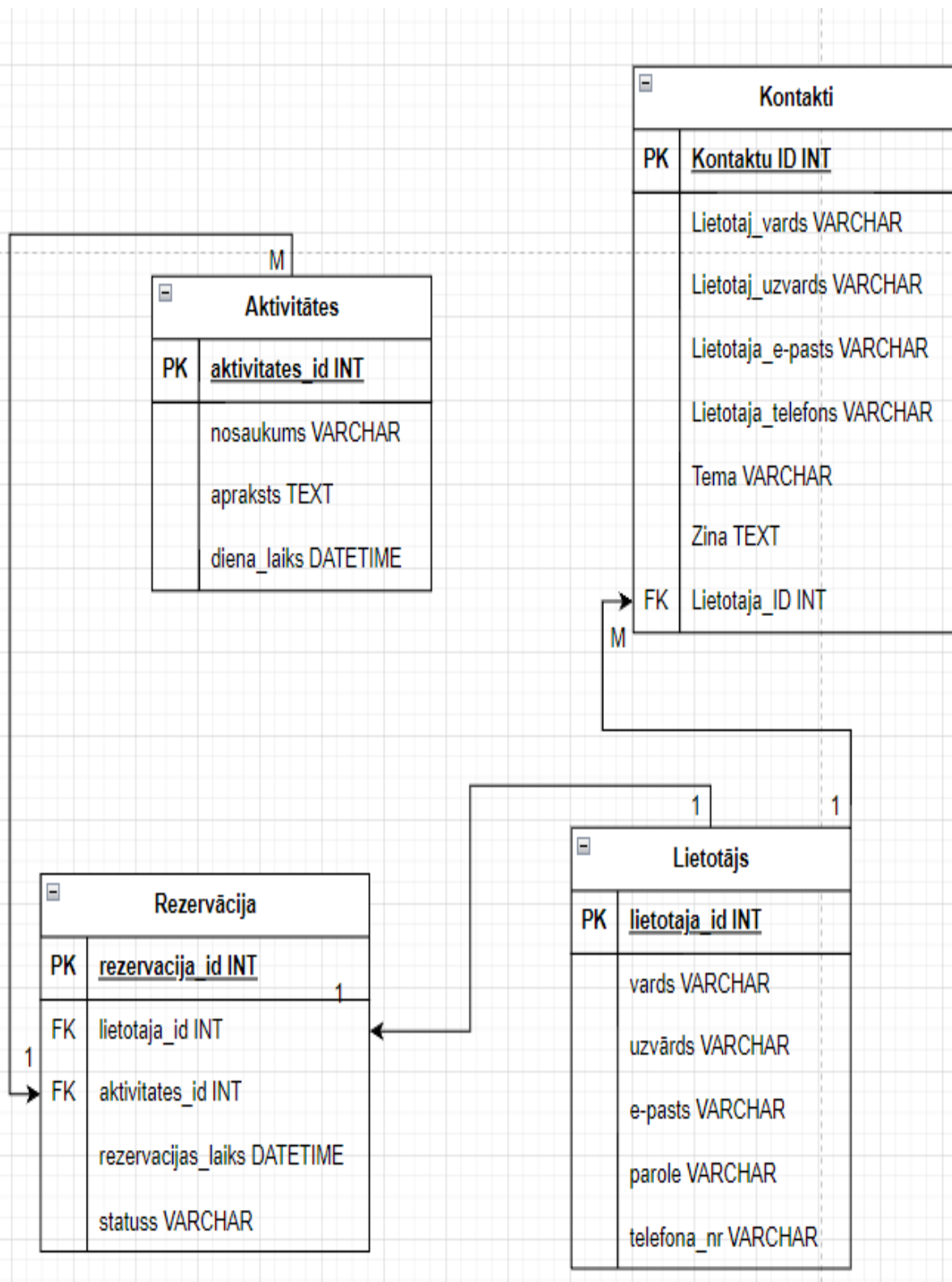
5.1.5. tabula.

5. Tabulas "Aktivitātes" struktūra

Nr.	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	aktivitātes_id	int	-	Aktivitātes primārā atslēga.
2.	nosaukums	varchar	255	Aktivitātes nosaukums.
3.	apraksts	text	-	Aktivitātes apraksts.
4.	atraššanās_vieta	varchar	255	Aktivitātes atrašanās vieta.
5.	datums_laiks	datetime	-	Aktivitātes datums un laiks.

Tabula "Aktivitātes" ir saistīta ar tabulām "Administrators" un "Rezervācija" caur laukiem admin_id un aktivitātes_id.

Datubāzes tabulu saišu shēma (skat. att. 5.1). Kā redzams datubāze sastāv no 5 tabulām, kas satur informāciju par lietotājiem, jaunumiem, kontaktiem, aktivitātēm un rezervāciju



6. LIETOTĀJA CEĻVEDIS

6.1. Sistēmas prasības

Lai veiksmīgi uzstādītu un lietotu "Brīvā laika pavadīšanas portālu", jūsu sistēmai ir jāatbilst sekojošām aparatūras un programmatūras prasībām.

Aparatūras prasības:

Procesors: Vismaz 2.0 GHz četrkodolu procesors vai jaudīgāks.

Operatīvā atmiņa (RAM): Vismaz 4 GB RAM, ieteicams 8 GB vai vairāk.

Cietais disks: Vismaz 20 GB brīvas vietas uz cietā diska.

Tīkls: Interneta pieslēgums ar minimālo ātrumu 1 Mbps.

Programmatūras prasības:

Operētājsistēma:

Windows 7 vai jaunāka versija

macOS 10.12 (Sierra) vai jaunāka versija

Linux (jebkura moderna izplatīšana)

Web serveris: Apache 2.4 vai jaunāka versija, Nginx vai cits savietojams web serveris.

PHP: PHP 7.4 vai jaunāka versija.

Datubāzes pārvaldības sistēma (DBVS): MySQL 5.7 vai jaunāka versija, MariaDB vai cits savietojams DBVS.

Papildus programmatūra:

phpMyAdmin 4.9 vai jaunāka versija (ieteicams, bet nav obligāts)

Composer (PHP atkarību pārvaldnieks)

Interneta pārlūkprogramma:

Google Chrome (vēlams pēdējā stabilā versija)

Mozilla Firefox (vēlams pēdējā stabilā versija)

Microsoft Edge (vēlams pēdējā stabilā versija)

Safari (vēlams pēdējā stabilā versija)

Papildus prasības:

Drošība: Pārliecinieties, ka serverī ir uzstādīts SSL sertifikāts, lai nodrošinātu šifrētu datu pārraidi.

Rezerves kopiju sistēma: Ieteicams uzstādīt automatizētu rezerves kopiju sistēmu, lai nodrošinātu datu drošību un integritāti.

6.2. Sistēmas instalācija un palaišana

Lai uzstādītu un palaistu "Brīvā laika pavadīšanas portālu", sekojiet šiem soļiem:

Priekšnosacījumi;

Pirms sākat instalāciju, pārliecinieties, ka jūsu sistēma atbilst sekojošām prasībām:

Web serveris (Apache 2.4 vai jaunāka versija, Nginx vai līdzīgs)

PHP 7.4 vai jaunāka versija

MySQL 5.7 vai jaunāka versija

Composer (PHP atkarību pārvaldnieks)

Interneta pārlūkprogramma (Google Chrome, Mozilla Firefox, Microsoft Edge vai Safari)

Lejupielādējiet programmas failus;

Lejupielādējiet programmas failus no jūsu norādītā avota (piemēram, GitHub repozitorija) un izpakojiet tos servera direktorijā.

```
# Klonējiet repozitoriju vai augšupielādējiet failus uz servera
git clone https://github.com/rvt-prog-kval-24/DP4-4-MatissSmits-BrivalaikaPavadisanasPortals.git /path/to/your/project
cd /path/to/your/project
```

Instalējiet atkarības;

Izmantojiet Composer, lai instalētu visas nepieciešamās PHP atkarības.

```
composer install
```

Izveidojiet datubāzi;

Pieslēdzieties MySQL serverim un izveidojiet datubāzi:

```
CREATE DATABASE briva_laika_pavadisanas_portals;
```

Importējiet sākotnējos datubāzes datus:

```
mysql -u jūsu_lietotājs -p briva_laika_pavadisanas_portals < database/initial_data.sql
```

Konfigurējiet vidi;

Kopējiet .env.example failu uz .env un aizpildiet nepieciešamos konfigurācijas parametrus (datubāzes pieslēgšanās dati, e-pasta konfigurācija utt.).

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

Ģenerējiet lietotņu atslēgu;

Lai nodrošinātu aplikācijas drošību, ģenerējiet lietotņu atslēgu:

```
php artisan key:generate
```

Palaidiet migrācijas;

Izpildiet datubāzes migrācijas, lai izveidotu nepieciešamās tabulas datubāzē:

```
php artisan migrate
```

Palaidiet serveri;

Ja izmantojat lokālo izstrādes vidi, palaidiet iebūvēto PHP serveri:

```
php artisan serve
```

Jūsu aplikācija būs pieejama lokāli šajā URL: <http://localhost:8000>.

Konfigurējiet web serveri produkcijas vidē;

Ja instalējat aplikāciju produkcijas vidē, konfigurējiet Apache vai Nginx, lai norādītu uz jūsu projekta direktoriju. Piemēram, šeit ir vienkārša Nginx konfigurācija:

```
server {  
    listen 80;  
    server_name jūsu-domenes-vārds.com;  
    root /path/to/your/project/public;  
  
    index index.php index.html index.htm;  
  
    location / {  
        try_files $uri $uri/ /index.php?$query_string;  
    }  
  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
    }  
  
    location ~ /\.ht {  
        deny all;  
    }  
}
```

Restartējiet Nginx, lai piemērotu izmaiņas:

```
sudo systemctl restart nginx
```

Pieklūstiet portālam;

Atveriet pārlūkprogrammu un dodieties uz jūsu domēna vārdu vai lokālo URL, lai piekļūtu "Brīvā laika pavadīšanas portālam".

Ja sekojat šiem soļiem, jums vajadzētu būt veiksmīgi uzstādītai un palaistai "Brīvā laika pavadīšanas portāla" sistēmai. Ja rodas problēmas vai jautājumi, lūdzu, sazinieties ar tehnisko atbalstu.

6.3. Programmas apraksts

"Brīvā laika pavadīšanas portāls" ir tīmekļa lietojumprogramma, kas ļauj lietotājiem atrast un rezervēt dažādas brīvā laika aktivitātes. Šī sadaļa sniedz detalizētu programmas funkcionalitātes aprakstu un tās sastāvdaļu pārskatu.

Galvenās funkcijas

Lietotāju reģistrācija un autentifikācija:

Reģistrācija: Lietotāji var izveidot kontu, norādot savu vārdu, uzvārdu, e-pasta adresi un paroli. Reģistrācija nodrošina piekļuvi visām portāla funkcijām;

Pieteikšanās: Reģistrētie lietotāji var pieteikties, izmantojot e-pasta adresi un paroli. Pieteikšanās ir nepieciešama, lai veiktu rezervācijas un piekļūtu personīgajam profilam.

Lietotāja profils:

Profila skatīšana un rediģēšana: Lietotāji var apskatīt un atjaunināt savu profilu, ieskaitot kontaktinformāciju un paroli;

Rezervāciju pārvaldība: Lietotāji var apskatīt savas veiktās rezervācijas un tās atcelt, ja nepieciešams.

Aktivitāšu pārvaldība:

Aktivitāšu pievienošana: Administratori var pievienot jaunas aktivitātes, norādot nosaukumu, aprakstu, dienu un laiku;

Aktivitāšu rediģēšana: Administratori var rediģēt esošās aktivitātes, atjauninot informāciju par aktivitātēm;

Aktivitāšu dzēšana: Administratori var dzēst nevajadzīgas vai beigušās aktivitātes.

Rezervāciju veikšana un pārvaldība:

Rezervācijas veikšana: Lietotāji var rezervēt aktivitātes, aizpildot rezervācijas formu un izvēloties vēlamo aktivitāti;

Rezervāciju pārvaldība: Lietotāji var apskatīt savu rezervāciju vēsturi un atcelt rezervācijas, ja nepieciešams.

Kontaktu pārvaldība:

Kontakta formas aizpildīšana: Lietotāji var nosūtīt ziņojumus portāla administrācijai, aizpildot kontakta formu ar savu vārdu, e-pasta adresi un ziņojumu;

Ziņojumu pārvaldība: Administratori var apskatīt un atbildēt uz lietotāju ziņojumiem.

Programmas komponentes:

Front-end (lietotāja interfeiss):

HTML/CSS/PHP: Lietotāja interfeiss ir veidots, izmantojot HTML, CSS un PHP, lai nodrošinātu intuitīvu un lietotājam draudzīgu saskarni;

Responsīvs dizains: Saskaņā ir izstrādāta tā, lai tā būtu ērti lietojama gan uz datora, gan mobilajās ierīcēs.

Back-end (servera puse):

PHP: Servera puse ir veidota, izmantojot PHP, lai nodrošinātu funkcionālības izpildi un datu apstrādi;

MySQL: Datu glabāšanai un pārvaldībai tiek izmantota MySQL datubāze.

Datu bāze:

Lietotāji: Glabā informāciju par reģistrētajiem lietotājiem (lietotāja ID, vārds, uzvārds, e-pasts, parole, telefona numurs, adrese);

Administratori: Glabā informāciju par administratoriem (admin ID, vārds, uzvārds, e-pasts, parole);

Aktivitātes: Glabā informāciju par pieejamajām aktivitātēm (aktivitātes ID, nosaukums, apraksts, atrašanās vieta, datums un laiks, cena);

Rezervācijas: Glabā informāciju par veiktajām rezervācijām (rezervācijas ID, lietotāja ID, aktivitātes ID, rezervācijas datums, statuss);

Kontakti: Glabā informāciju par lietotāju ziņojumiem (kontakta ID, lietotāja ID, vārds, uzvārds, e-pasts, ziņojums, datums).

Drošība:

Portāls izmanto vairākus drošības pasākumus, lai aizsargātu lietotāju datus un nodrošinātu sistēmas drošību:

Paroļu šifrēšana: Lietotāju paroles tiek glabātas šifrētā veidā, izmantojot drošu šifrēšanas algoritmu;

SSL/TLS: Datu pārraide starp lietotājiem un serveri tiek šifrēta, izmantojot SSL/TLS protokolu;

Autentifikācija un autorizācija: Tikai autorizēti lietotāji un administratori var piekļūt attiecīgajām funkcijām un datiem.

6.4. Testa piemērs

Zemāk tabulā (skat. Att. 6.4) varēsiet aplukot testa piemērus, Jauna lietotāja registrācija; lietotājvārds, lietotāja e-pasts, parole, atkārtotā parole. Aktivitātes pievienošana un rediģēšana; administrators ievada aktivitātes datus un rediģē aktivitātes datus. Rezervāciju veikšana un dzēšana; lietotājs izvēlas aktivitāti un veic rezervāciju un lietotājs dzēš rezervāciju. Kontakta veidlapas aizpildīšana; Lietotājs aizpilda un iesniedz kontakta veidlapu.

Nr.	Prasības numurs	Prasības nosaukums	Ievaddati/situācijas apraksts	Sagaidāmais rezultāts	Statuss
1.	1.1	Jauna lietotāja reģistrācija	Pareizi dati: lietotāja vārds, lietotāja e-pasts, parole, parole	Datu bāzē tiek pievienots ieraksts, saskarnē parādās paziņojums par veiksmīgu reģistrāciju	Pareizi
2.	1.2	Jauna lietotāja reģistrācija	Lietotāja vārds = peterisA	Paziņojums par jau eksistējošo lietotāju	Pareizi
3.	1.3	Jauna lietotāja reģistrācija	Lietotāja e-pasts = peteris@gmail.com	Paziņojums par lietotāja e-pasta nepareizo formātu	Pareizi
4.	2.1	Aktivitātes pievienošana	Administrators ievada aktivitātes datus	Aktivitāte tiek pievienota datu bāzei, saskarnē parādās paziņojums "Aktivitāte pievienota"	Pareizi
5.	2.2	Aktivitātes rediģēšana	Administrators rediģē aktivitātes datus	Aktivitātes dati tiek atjaunināti datu bāzē, saskarnē parādās paziņojums "Aktivitāte atjaunināta"	Pareizi
6.	3.1	Rezervācijas veikšana	Lietotājs izvēlas aktivitāti un veic rezervāciju	Rezervācija tiek pievienota datu bāzei, saskarnē parādās paziņojums "Rezervācija veikta"	Pareizi
7.	3.2	Rezervācijas dzēšana	Lietotājs dzēš rezervāciju	Rezervācija tiek dzēsta no datu bāzes, saskarnē parādās paziņojums "Rezervācija dzēsta"	Pareizi
8.	4.1	Kontakta veidlapas aizpildīšana	Lietotājs aizpilda un iesniedz kontakta veidlapu	Kontakta dati tiek pievienoti datu bāzei, saskarnē parādās paziņojums "Ziņojums nosūtīts"	Pareizi

6.4. att. Testa piemēra tabula.

NOBEIGUMS

"Brīvā laika pavadīšanas portāls" ir izstrādāts, lai piedāvātu lietotājiem ērtu un efektīvu veidu, kā plānot un rezervēt dažādas brīvā laika aktivitātes. Portāls nodrošina plašu funkcionalitāti, kas ietver lietotāju reģistrāciju un autentifikāciju, profila pārvaldību, aktivitāšu pievienošanu un rediģēšanu, rezervāciju veikšanu un atcelšanu, kā arī kontaktu ziņojumu pārvaldību.

Galvenie sasniegumi:

Lietotāju draudzīgums:

Izstrādāta intuitīva un ērti lietojama saskarne, kas nodrošina patīkamu lietotāja pieredzi.

Responsīvs dizains, kas pielāgojas dažādiem ekrānu izmēriem un ierīcēm.

Efektīva pārvaldība:

Administratoriem ir iespēja efektīvi pārvaldīt aktivitātes un lietotāju rezervācijas.

Sistēma nodrošina ātru un vienkāršu datu ievadi un atjaunināšanu.

Drošība:

Portāls izmanto vairākus drošības pasākumus, lai aizsargātu lietotāju datus, tostarp paroļu šifrēšanu un SSL/TLS protokolu datu pārraidei.

Tikai autorizēti lietotāji un administratori var piekļūt noteiktām funkcijām un datiem.

Tehnoloģiskā stabilitāte:

Izmantotas modernas un uzticamas tehnoloģijas, tostarp PHP, MySQL un Composer, lai nodrošinātu stabilu un drošu sistēmas darbību.

Veikta rūpīga testēšana, lai identificētu un novērstu iespējamās kļūdas un trūkumus.

Projekta izstrādes procesā tika ņemti vērā visi galvenie prasību specifikācijā norādītie nosacījumi, kā arī nodrošināta pilna funkcionalitāte saskaņā ar plānoto sistēmas arhitektūru. Veiksmīgi izstrādāta datubāzes struktūra un integrētas visas nepieciešamās tabulas un to saistības.

Turpmākie uzlabojumi

Papildu funkcionalitāte:

Izveidot atsauksmju un vērtējumu sistēmu lietotājiem, lai tie varētu novērtēt aktivitātes.

Ieviest maksājumu integrāciju, lai lietotāji varētu veikt tiešsaistes maksājumus par rezervācijām.

Veiktspējas uzlabošana:

Optimizēt datubāzes vaicājumus un sistēmas darbību, lai nodrošinātu ātrāku atbildes laiku un labāku lietotāja pieredzi.

Izmantot kešošanas tehnoloģijas, lai samazinātu servera noslodzi un uzlabotu lapu ielādes ātrumu.

Lietotāja pieredzes uzlabošana:

Veikt regulāras lietotāja interfeisa uzlabošanas, pamatojoties uz lietotāju atsauksmēm un testēšanas rezultātiem.

Organizēt lietotāju apmācības un sniegt atbalstu, lai palīdzētu tiem efektīvi izmantot portāla funkcionalitāti.

Projekta mērķis ir nodrošināt, ka "Brīvā laika pavadīšanas portāls" kļūst par uzticamu un populāru rīku, kas palīdz lietotājiem plānot un izbaudīt savu brīvo laiku. Ar regulāru uzturēšanu un uzlabošanu šis portāls spēs apmierināt lietotāju vajadzības un nodrošināt augstu pakalpojumu kvalitāti.

Piebilde:

Viss ieplānotais tika sasniegts, varbūt ar ne tik lielu precizitāti, bet pa lielam viss ir gatavs kā ieplānoju, protams ir kur pielikt roku, bet galvenais ka esmu kaut cik apmierināts ar to kas man ir sanācis šīs programmas izveides beigās.

Šo projektu veidojot nācās saskarties ar grūtībām un neveiksmēm, piemēram, nācās visus html failus pārveidot par php failiem lai labāk tie darbotos ar datubāzēm, jo veidoju datubāzes caur MYSQL(MYPHPADMIN) weblapu.

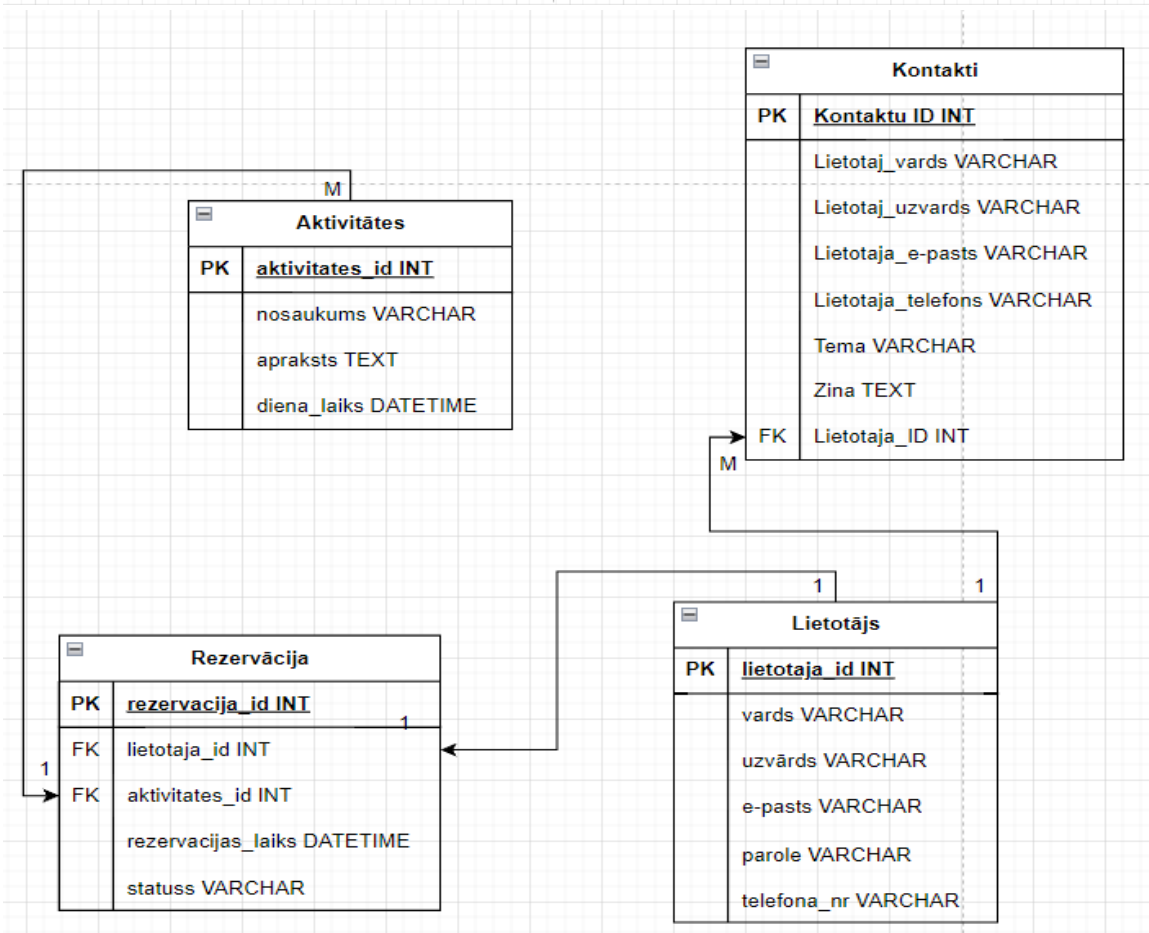
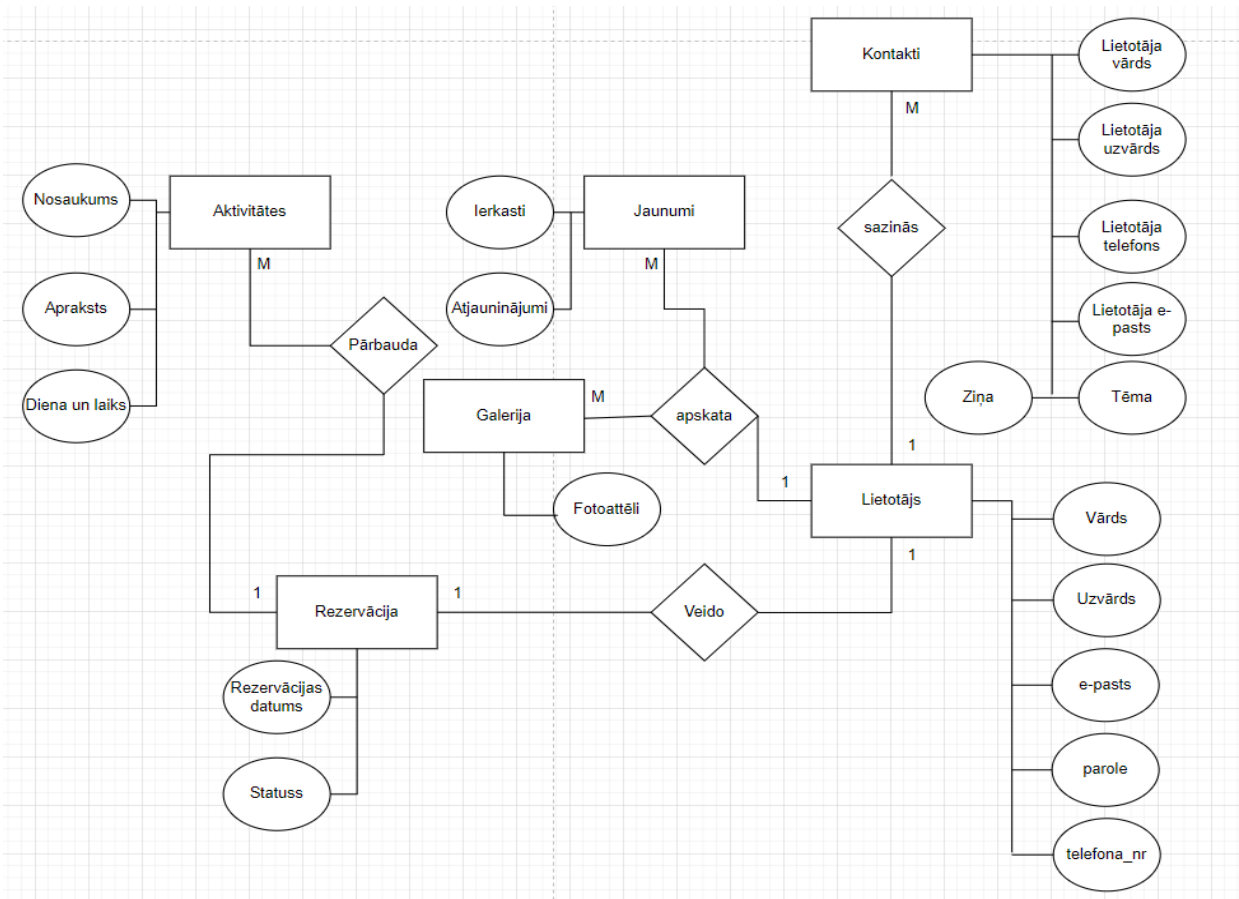
Toties viss ir sanācis, plānoju turpmāk iespējams nodarboties ar šo projektu vēl padziļinātāk lai pilnībā pilnveidotu un nebūtu, kā acīs netīrums kuru prasās izvilkt ārā, par projektu runājot, acīs metās lietas kuras prasās pielabot, jeb pārtaisīt uz ko labāku un lai acīs tas tā vairāk nemetās.

INFORMĀCIJAS AVOTI

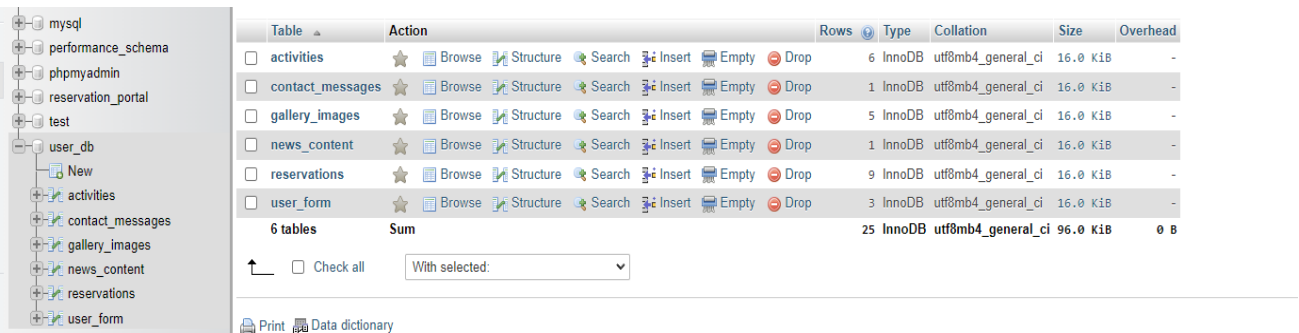
1. Datubāžu izveidošanas portāls izmantojot PHP un MYSQL(01.06.24)
<https://www.phpmyadmin.net/>
2. HTML lapas basic izveidošana, izmantojot navigācijas(pārmešana no lapas lapā)(01.06.24)<https://www.w3schools.com/html/default.asp>
3. Mājaslapas dizaina izveidošana palīg rīku meklēšana izmantojot šo saiti informācijai(01.06.24)https://www.w3schools.com/html/html_css.asp
4. PHP dažādu funkciju aplūkošana piemēram echo print(01.06.24)
https://www.w3schools.com/php/php_echo_print.asp
5. PHP IF un ELSE funkcijas izmantošana, piemēram, datubāzēs kad veicu ierkastus, ja izmet kodu nostrādā ELSE kods, bet IF kad veicu ierakstu un viņš parāda izmaiņas.(01.06.24) https://www.w3schools.com/php/php_if_else.asp
6. Paraugs pēc kā orientējos ER modeļa diagrammas veidošanā(01.06.24)
https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model
7. UML diagrammas veidošanas paraugs(01.06.24)
https://en.wikipedia.org/wiki/SQL#SQL_data_types
8. Dokumentācijas dizaina uzlabošana(08.11.23)
<https://e.rvt.lv/mod/resource/view.php?id=19115>

PIELIKUMI

Pielikums 1. Pārskatu piemēri ER-diagramma un Datubāzes tabulu saišu shēma:



Pielikums 2. Datubāzes ekrānattēlu piemēri;

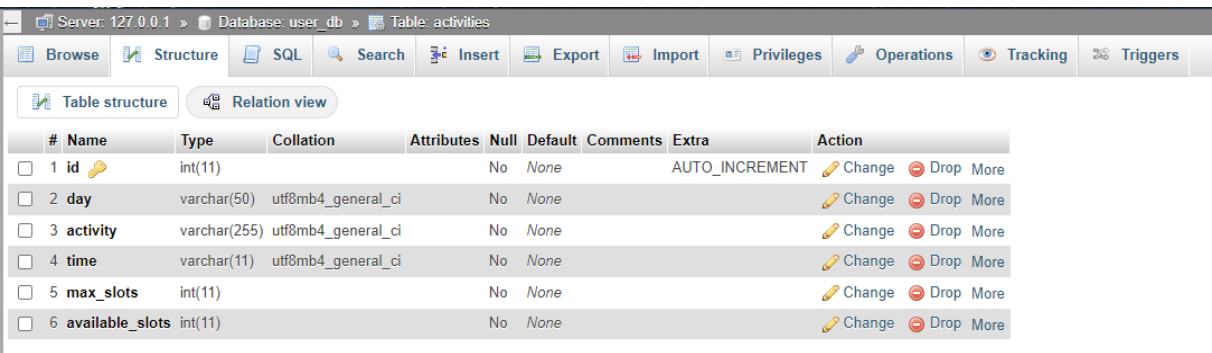


The screenshot shows the phpMyAdmin interface. On the left, a tree view shows the database structure: mysql, performance_schema, phpmyadmin, reservation_portal, test, and user_db. The 'user_db' database is selected, and its tables are listed: activities, contact_messages, gallery_images, news_content, reservations, and user_form. The main panel displays a table summary for 'user_db' with 6 tables. The summary table is as follows:

Table	Action	Rows	Type	Collation	Size	Overhead
activities	[Star] [Browse] [Structure] [Search] [Insert] [Empty] [Drop]	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
contact_messages	[Star] [Browse] [Structure] [Search] [Insert] [Empty] [Drop]	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
gallery_images	[Star] [Browse] [Structure] [Search] [Insert] [Empty] [Drop]	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
news_content	[Star] [Browse] [Structure] [Search] [Insert] [Empty] [Drop]	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
reservations	[Star] [Browse] [Structure] [Search] [Insert] [Empty] [Drop]	9	InnoDB	utf8mb4_general_ci	16.0 KiB	-
user_form	[Star] [Browse] [Structure] [Search] [Insert] [Empty] [Drop]	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
6 tables	Sum	25	InnoDB	utf8mb4_general_ci	96.0 KiB	0 B

Below the summary table, there are checkboxes for 'Check all' and a dropdown menu for 'With selected:'. At the bottom, there are links for 'Print' and 'Data dictionary'.

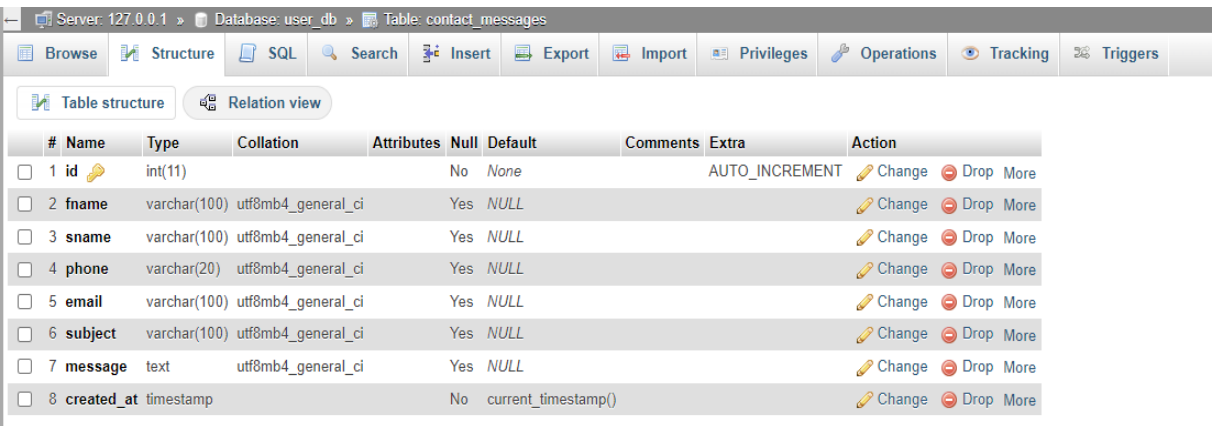
Tabulas aktivitātes struktūra:



The screenshot shows the 'Table structure' view for the 'activities' table in the 'user_db' database. The table has 6 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	[Change] [Drop] [More]
2	day	varchar(50)	utf8mb4_general_ci		No	None			[Change] [Drop] [More]
3	activity	varchar(255)	utf8mb4_general_ci		No	None			[Change] [Drop] [More]
4	time	varchar(11)	utf8mb4_general_ci		No	None			[Change] [Drop] [More]
5	max_slots	int(11)			No	None			[Change] [Drop] [More]
6	available_slots	int(11)			No	None			[Change] [Drop] [More]

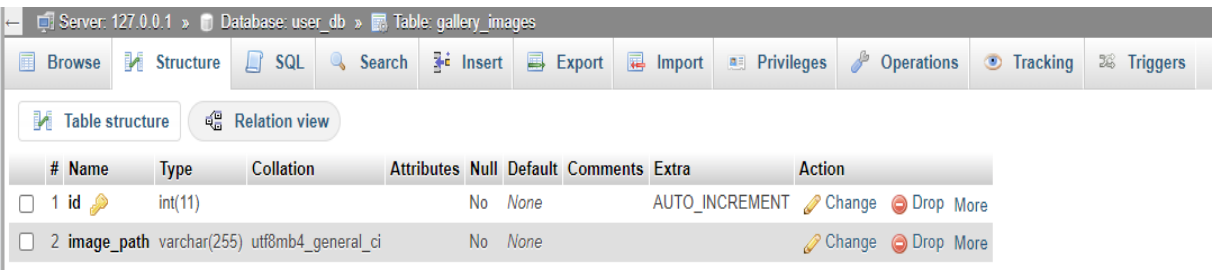
Tabulas kontakti struktūra:



The screenshot shows the 'Table structure' view for the 'contact_messages' table in the 'user_db' database. The table has 8 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	[Change] [Drop] [More]
2	fname	varchar(100)	utf8mb4_general_ci		Yes	NULL			[Change] [Drop] [More]
3	sname	varchar(100)	utf8mb4_general_ci		Yes	NULL			[Change] [Drop] [More]
4	phone	varchar(20)	utf8mb4_general_ci		Yes	NULL			[Change] [Drop] [More]
5	email	varchar(100)	utf8mb4_general_ci		Yes	NULL			[Change] [Drop] [More]
6	subject	varchar(100)	utf8mb4_general_ci		Yes	NULL			[Change] [Drop] [More]
7	message	text	utf8mb4_general_ci		Yes	NULL			[Change] [Drop] [More]
8	created_at	timestamp			No	current_timestamp()			[Change] [Drop] [More]

Tabulas galerijas struktūra:



The screenshot shows the 'Table structure' view for the 'gallery_images' table in the 'user_db' database. The table has 2 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	[Change] [Drop] [More]
2	image_path	varchar(255)	utf8mb4_general_ci		No	None			[Change] [Drop] [More]

Tabulas jaunumi struktūra:

Server: 127.0.0.1 » Database: user_db » Table: news_content

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Tracking](#)
[Triggers](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	content	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

Tabulas rezervācijas struktūra:

Server: 127.0.0.1 » Database: user_db » Table: reservations

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Tracking](#)
[Triggers](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	user_id	int(11)			No	None			Change Drop More
<input type="checkbox"/> 3	activity_id	int(11)			No	None			Change Drop More
<input type="checkbox"/> 4	reservation_date	date			No	None			Change Drop More
<input type="checkbox"/> 5	message	text	utf8mb4_general_ci		No	None			Change Drop More

Tabulas lietotāji struktūra:

Server: 127.0.0.1 » Database: user_db » Table: user_form

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Tracking](#)
[Triggers](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(255)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	email	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 5	user_type	varchar(255)	utf8mb4_general_ci		No	user			Change Drop More
<input type="checkbox"/> 6	surname	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 7	phone	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Pielikums 3. Programmas pirmsteksta kods AdminLapa visas datubāzes saistīšana un programmā visu datubāžu rediģēšana:

```
<?php
```

```
@include 'config.php';
```

```
session_start();
```

```
if (!isset($_SESSION['user_name']) || $_SESSION['user_type'] !== 'admin') {
```

```
    header('Location: login_form.php');
```

```
    exit();
```

```
}
```

```
// Add activity
```

```
if (isset($_POST['add_activity'])) {
```

```
    $day = mysqli_real_escape_string($conn, $_POST['day']);
```

```
    $activity = mysqli_real_escape_string($conn, $_POST['activity']);
```

```
    $time = mysqli_real_escape_string($conn, $_POST['time']);
```

```
    $max_slots = (int) $_POST['max_slots'];
```

```
    $insert = "INSERT INTO activities (day, activity, time, max_slots, available_slots)
VALUES ('$day', '$activity', '$time', $max_slots, $max_slots)";
```

```
if (mysqli_query($conn, $insert)) {
```

```
    $message = "Aktivitāte pievienota veiksmīgi!";
```

```
} else {
```

```
    $error = "Kļūda pievienojot aktivitāti!";
```

```
}
```

```
}
```

```
// Handle message deletion
```

```
if (isset($_POST['delete_message'])) {
```

```
    $message_id = mysqli_real_escape_string($conn, $_POST['message_id']);
```

```

$delete_query = "DELETE FROM contact_messages WHERE id = $message_id";
if (mysqli_query($conn, $delete_query)) {
    $message = "Ziņojums dzēsts veiksmīgi!";
} else {
    $error = "Kļūda dzēšot ziņojumu!";
}
}

// Fetch news content
$news_query = "SELECT * FROM news_content ORDER BY id ASC LIMIT 1";
$news_result = mysqli_query($conn, $news_query);
$news = mysqli_fetch_assoc($news_result);

// Update news content
if (isset($_POST['edit_news'])) {
    $news_content = mysqli_real_escape_string($conn, $_POST['news_content']);

    $update_news = "UPDATE news_content SET content = '$news_content' WHERE id = " .
    $news['id'];

    if (mysqli_query($conn, $update_news)) {
        $message = "Jaunumu saturs veiksmīgi atjaunināts!";
    } else {
        $error = "Kļūda atjauninot jaunumu saturu!";
    }
}

// Edit activity
if (isset($_POST['edit_activity'])) {

```

```

$id = mysqli_real_escape_string($conn, $_POST['id']);
$day = mysqli_real_escape_string($conn, $_POST['day']);
$activity = mysqli_real_escape_string($conn, $_POST['activity']);
$time = mysqli_real_escape_string($conn, $_POST['time']);
$max_slots = (int) $_POST['max_slots'];

$update = "UPDATE activities SET day = '$day', activity = '$activity', time = '$time',
max_slots = $max_slots, available_slots = $max_slots WHERE id = $id";

if (mysqli_query($conn, $update)) {
    $message = "Aktivitāte atjaunināta veiksmīgi!";
} else {
    $error = "Kļūda atjauninot aktivitāti!";
}
}

// Filter activities
$filter_day = isset($_POST['filter_day']) ? $_POST['filter_day'] : "";
$filter_activity = isset($_POST['filter_activity']) ? $_POST['filter_activity'] : "";

$filter_query = "SELECT * FROM activities";
$conditions = [];

if ($filter_day != "") {
    $conditions[] = "day = '$filter_day'";
}

if ($filter_activity != "") {
    $conditions[] = "activity LIKE '%$filter_activity%'";
}

```

```

if (count($conditions) > 0) {
    $filter_query .= " WHERE " . implode(' AND ', $conditions);
}

$filter_query .= " ORDER BY FIELD(day, 'Pirmdiena', 'Otradiena', 'Trešdiena', 'Ceturtdiena',
'Piektdiena', 'Sestdiena', 'Svētdiena'), time ASC";

$activities_result = mysqli_query($conn, $filter_query);

$days = ['Pirmdiena', 'Otradiena', 'Trešdiena', 'Ceturtdiena', 'Piektdiena', 'Sestdiena',
'Svētdiena'];

// Fetch users
$users_query = "SELECT * FROM user_form";
$users_result = mysqli_query($conn, $users_query);

// Fetch reservations
$filter_reservation_day = isset($_POST['filter_reservation_day']) ?
$_POST['filter_reservation_day'] : "";

$filter_reservation_name = isset($_POST['filter_reservation_name']) ?
$_POST['filter_reservation_name'] : "";

$filter_reservation_surname = isset($_POST['filter_reservation_surname']) ?
$_POST['filter_reservation_surname'] : "";

$filter_reservation_activity = isset($_POST['filter_reservation_activity']) ?
$_POST['filter_reservation_activity'] : "";

$reservation_filter_query = "SELECT reservations.*, user_form.name, user_form.surname,
activities.activity, activities.day
        FROM reservations
        JOIN user_form ON reservations.user_id = user_form.id
        JOIN activities ON reservations.activity_id = activities.id";

$reservation_conditions = [];

```



```

if ($filter_reservation_day != "") {
    $reservation_conditions[] = "activities.day = '$filter_reservation_day'";
}

if ($filter_reservation_name != "") {
    $reservation_conditions[] = "user_form.name LIKE '%$filter_reservation_name%'";
}

if ($filter_reservation_surname != "") {
    $reservation_conditions[] = "user_form.surname LIKE '%$filter_reservation_surname%'";
}

if ($filter_reservation_activity != "") {
    $reservation_conditions[] = "activities.activity LIKE '%$filter_reservation_activity%'";
}

if (count($reservation_conditions) > 0) {
    $reservation_filter_query .= " WHERE " . implode(' AND ', $reservation_conditions);
}

$reservation_filter_query .= " ORDER BY activities.day, activities.time";
$reservations_result = mysqli_query($conn, $reservation_filter_query);

// Edit user
if (isset($_POST['edit_user'])) {
    $id = mysqli_real_escape_string($conn, $_POST['id']);
    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $surname = mysqli_real_escape_string($conn, $_POST['surname']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);

```

```
$phone = mysqli_real_escape_string($conn, $_POST['phone']);
```

```
$user_type = mysqli_real_escape_string($conn, $_POST['user_type']);
```

```
$update = "UPDATE user_form SET name = '$name', surname = '$surname', email = '$email', phone = '$phone', user_type = '$user_type' WHERE id = $id";
```

```
if (mysqli_query($conn, $update)) {
```

```
    $message = "Lietotājs atjaunināts veiksmīgi!";
```

```
} else {
```

```
    $error = "Kļūda atjauninot lietotāju!";
```

```
}
```

```
}
```

```
// Delete user
```

```
if (isset($_POST['delete_user'])) {
```

```
    $id = mysqli_real_escape_string($conn, $_POST['id']);
```

```
$delete = "DELETE FROM user_form WHERE id = $id";
```

```
if (mysqli_query($conn, $delete)) {
```

```
    $message = "Lietotājs dzēsts veiksmīgi!";
```

```
} else {
```

```
    $error = "Kļūda dzēšot lietotāju!";
```

```
}
```

```
}
```

```
// Filter users
```

```
$filter_name = isset($_POST['filter_name']) ? $_POST['filter_name'] : "";
```

```
$filter_email = isset($_POST['filter_email']) ? $_POST['filter_email'] : "";
```

```
$filter_user_type = isset($_POST['filter_user_type']) ? $_POST['filter_user_type'] : "";
```

```

$user_filter_query = "SELECT * FROM user_form";

$user_conditions = [];

if ($filter_name != "") {
    $user_conditions[] = "name LIKE '%$filter_name%'";
}

if ($filter_email != "") {
    $user_conditions[] = "email LIKE '%$filter_email%'";
}

if ($filter_user_type != "") {
    $user_conditions[] = "user_type = '$filter_user_type'";
}

if (count($user_conditions) > 0) {
    $user_filter_query .= " WHERE " . implode(' AND ', $user_conditions);
}

$user_filter_query .= " ORDER BY id ASC";

$users_result = mysqli_query($conn, $user_filter_query);

if (isset($_POST['upload_image'])) {
    $target_dir = "uploads/";

    $target_file = $target_dir . basename($_FILES["image"]["name"]);

    $uploadOk = 1;

    $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));

    // Check if image file is a actual image or fake image

```

```
$check = getimagesize($_FILES["image"]["tmp_name"]);  
if ($check !== false) {  
    $uploadOk = 1;  
} else {  
    $error = "File is not an image.";  
    $uploadOk = 0;  
}  
  
// Check if file already exists  
if (file_exists($target_file)) {  
    $error = "Sorry, file already exists.";  
    $uploadOk = 0;  
}  
  
// Check file size  
if ($_FILES["image"]["size"] > 500000) {  
    $error = "Sorry, your file is too large.";  
    $uploadOk = 0;  
}  
  
// Allow certain file formats  
if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"  
&& $imageFileType != "gif") {  
    $error = "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";  
    $uploadOk = 0;  
}  
  
// Check if $uploadOk is set to 0 by an error  
if ($uploadOk == 0) {  
    // Handle error
```

```

} else {

    if (move_uploaded_file($_FILES["image"]["tmp_name"], $target_file)) {

        $insert = "INSERT INTO gallery_images (image_path) VALUES ('$target_file')";

        if (mysqli_query($conn, $insert)) {

            $message = "Image uploaded successfully!";

        } else {

            $error = "Error uploading image!";

        }

    } else {

        $error = "Sorry, there was an error uploading your file.";

    }

}

}

```

// Handle image deletion

```

if (isset($_POST['delete_image'])) {

    $image_id = $_POST['image_id'];

    $query = "SELECT image_path FROM gallery_images WHERE id = $image_id";

    $result = mysqli_query($conn, $query);

    if ($row = mysqli_fetch_assoc($result)) {

        $image_path = $row['image_path'];

        if (unlink($image_path)) {

            $delete_query = "DELETE FROM gallery_images WHERE id = $image_id";

            if (mysqli_query($conn, $delete_query)) {

                $message = "Image deleted successfully!";

            } else {

                $error = "Error deleting image from database!";

            }

        } else {

            $error = "Error deleting image file!";

        }

    }

}

```

```

    }
}
}

// Fetch images for gallery
$gallery_query = "SELECT * FROM gallery_images";
$gallery_result = mysqli_query($conn, $gallery_query);
?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <meta name="description" content="BrivaLaikaPavadisanasPortals, Latvija">

    <meta name="keywords" content="BrivaLaikaPavadisanasPortals, gym, Spelulaukums,
aktivitates, latvija">

    <link rel="stylesheet" href="assets/css/style1.css">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link
href="https://fonts.googleapis.com/css2?family=Libre+Bodoni:ital@1&family=Montserrat:wght@300&family=Open+Sans:ital@1&display=swap" rel="stylesheet">

    <title>Admin Page</title>

    <!-- Favicon -->

    <link rel="apple-touch-icon" sizes="180x180" href="assets/images/apple-touch-icon.png">

    <link rel="icon" type="image/png" sizes="32x32" href="assets/images/favicon-32x32.png">

```

```
<link rel="icon" type="image/png" sizes="16x16" href="assets/images/favicon-16x16.png">
```

```
<script  
src="https://cdn.tiny.cloud/1/psaegtq7q15xsvtrto6au8rkzs8elaf8nkr2lr0thulozmr7/tinymce/7/tinymce.min.js" referrerpolicy="origin"></script>
```

```
<script>
```

```
tinymce.init({
```

```
  selector: 'textarea',
```

```
  plugins: 'anchor autolink charmap codesample emoticons image link lists media  
searchreplace table visualblocks wordcount checklist mediaembed casechange export  
formatpainter pageembed linkchecker a11ychecker tinymcespellchecker permanentpen  
powerpaste advtable advcode editimage advtemplate ai mentions tinycomments  
tableofcontents footnotes mergetags autocorrect typography inlinecss markdown',
```

```
  toolbar: 'undo redo | blocks fontfamily fontsize | bold italic underline strikethrough | link  
image media table mergetags | addcomment showcomments | spellcheckdialog a11ycheck  
typography | align lineheight | checklist numlist bullist indent outdent | emoticons charmap |  
removeformat',
```

```
  tinycomments_mode: 'embedded',
```

```
  tinycomments_author: 'Author name',
```

```
  mergetags_list: [
```

```
    { value: 'First.Name', title: 'First Name' },
```

```
    { value: 'Email', title: 'Email' },
```

```
  ],
```

```
  ai_request: (request, respondWith) => respondWith.string(() => Promise.reject("See docs  
to implement AI Assistant")),
```

```
});
```

```
</script>
```

```
<style>
```

```
  body {
```

```
    font-family: 'Montserrat', sans-serif;
```

```
    margin: 0;
```

```
    padding: 0;
```

```
    box-sizing: border-box;

    background: url('assets/images/background.jpg') no-repeat center center fixed;

    background-size: cover;

    color: #333;
}
```

```
header {

    background: rgba(255, 255, 255, 0.8);

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
```

```
#hero {

    text-align: center;

    padding: 20px 0;
}
```

```
#logo {

    font-family: 'Libre Bodoni', serif;

    font-size: 2rem;

    color: #5a8f7b;

    animation: fadeInDown 1s ease-in-out;
}
```

```
nav {

    display: flex;

    justify-content: center;

    background: #5a8f7b;
}
```

```
nav ul {
```



```
list-style: none;
padding: 0;
margin: 0;
display: flex;
}
```

```
nav li {
    margin: 0 15px;
}
```

```
nav a {
    color: #fff;
    text-decoration: none;
    font-size: 1rem;
    padding: 10px 15px;
    display: block;
    transition: background 0.3s ease;
}
```

```
nav a:hover,
.active-page {
    background: #457c63;
    border-radius: 5px;
}
```

```
.container {
    display: flex;
    justify-content: center;
    margin: 60px 20px;
    flex-wrap: wrap;
}
```

```
}
```

```
.sidebar {  
  width: 250px;  
  background: #5a8f7b;  
  color: #fff;  
  padding: 20px;  
  border-radius: 10px;  
  margin: 20px;  
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}
```

```
.sidebar h3 {  
  font-size: 1.2rem;  
  margin-bottom: 10px;  
}
```

```
.sidebar ul {  
  list-style: none;  
  padding: 0;  
}
```

```
.sidebar li {  
  margin: 10px 0;  
}
```

```
.sidebar a {  
  color: #fff;  
  text-decoration: none;  
  display: block;
```

```
padding: 10px;
border-radius: 5px;
transition: background 0.3s ease;
}
```

```
.sidebar a:hover {
    background: #457c63;
}
```

```
.content {
    flex: 1;
    background: rgba(255, 255, 255, 0.9);
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    margin: 20px;
    max-width: 800px;
    overflow-x: auto;
}
```

```
.content h2 {
    margin-bottom: 20px;
    color: #5a8f7b;
}
```

```
form {
    background: rgba(255, 255, 255, 0.9);
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
```

```
    text-align: center;
    margin-bottom: 20px;
}
```

```
form h3 {
    margin-bottom: 20px;
    color: #5a8f7b;
}
```

```
form .fill-out {
    width: calc(100% - 20px);
    padding: 10px;
    margin: 10px 0;
    border-radius: 5px;
    border: 1px solid #ccc;
}
```

```
form .form-btn {
    padding: 10px 20px;
    background: #5a8f7b;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background 0.3s ease;
    width: 100%;
}
```

```
form .form-btn:hover {
    background: #457c63;
```

```
}
```

```
.activities-container,
```

```
.users-container,
```

```
.reservations-container {
```

```
    background: rgba(255, 255, 255, 0.9);
```

```
    padding: 20px;
```

```
    border-radius: 10px;
```

```
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
```

```
    margin-top: 20px;
```

```
    overflow-x: auto;
```

```
}
```

```
.activities-table,
```

```
.users-table,
```

```
.reservations-table,
```

```
.gallery-table {
```

```
    width: 100%;
```

```
    border-collapse: collapse;
```

```
}
```

```
.activities-table th, .activities-table td,
```

```
.users-table th, .users-table td,
```

```
.reservations-table th, .reservations-table td,
```

```
.gallery-table th, .gallery-table td {
```

```
    border: 1px solid #ddd;
```

```
    padding: 8px;
```

```
    text-align: left;
```

```
}
```

```
.activities-table th, .users-table th, .reservations-table th, .gallery-table th {  
    background-color: #5a8f7b;  
    color: white;  
}
```

```
.activities-table td a,  
.users-table td a,  
.reservations-table td a,  
.gallery-table td a {  
    margin-right: 10px;  
    color: #5a8f7b;  
    text-decoration: none;  
}
```

```
.activities-table td a:hover,  
.users-table td a:hover,  
.reservations-table td a:hover,  
.gallery-table td a:hover {  
    text-decoration: underline;  
}
```

```
.error-msg {  
    color: red;  
    font-size: 14px;  
    margin-bottom: 10px;  
}
```

```
.success-msg {  
    color: green;  
    font-size: 14px;
```

```
margin-bottom: 10px;  
}
```

```
/* Modal Styles */
```

```
.modal {  
    display: none;  
    position: fixed;  
    z-index: 1;  
    left: 0;  
    top: 0;  
    width: 100%;  
    height: 100%;  
    overflow: auto;  
    background-color: rgb(0, 0, 0);  
    background-color: rgba(0, 0, 0, 0.4);  
    padding-top: 60px;  
}
```

```
.modal-content {  
    background-color: #fefefe;  
    margin: 5% auto;  
    padding: 20px;  
    border: 1px solid #888;  
    width: 80%;  
    max-width: 500px;  
    border-radius: 10px;  
}
```

```
.close {  
    color: #aaa;
```

```
float: right;

font-size: 28px;

font-weight: bold;

}
```

```
.close:hover,

.close:focus {

    color: black;

    text-decoration: none;

    cursor: pointer;

}
```

```
footer {

    text-align: center;

    background: #5a8f7b;

    color: #fff;

    padding: 20px;

    position: relative;

    bottom: 0;

    box-shadow: 0 -4px 8px rgba(0, 0, 0, 0.1);

    margin-top: 40px;

}
```

```
footer ul {

    list-style: none;

    padding: 0;

    margin: 20px 0 10px;

    display: flex;

    justify-content: center;

}
```



```
footer li {  
  margin: 0 10px;  
}
```

```
footer a {  
  color: #fff;  
  text-decoration: none;  
  font-size: 1.5rem;  
}
```

```
footer a:hover {  
  color: #ddd;  
}
```

```
.filter-sidebar {  
  width: 250px;  
  background: #5a8f7b;  
  color: #fff;  
  padding: 20px;  
  border-radius: 10px;  
  margin: 20px;  
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
  flex-shrink: 0;  
}
```

```
.filter-sidebar h3 {  
  font-size: 1.2rem;  
  margin-bottom: 10px;  
}
```

```
.filter-sidebar form {  
    display: flex;  
    flex-direction: column;  
}
```

```
.filter-sidebar form select,  
.filter-sidebar form input,  
.filter-sidebar form .form-btn {  
    margin: 10px 0;  
}
```

```
.main-content {  
    display: flex;  
    width: 100%;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<!--Logo, menu linki-->
```

```
<header>
```

```
<div id="hero">
```

```
    <h1 id="logo"><i class="fa-solid fa-tree"></i> Briva Laika Pavadisanas Portals <i  
class="fa-solid fa-person-walking"></i>
```

```
    </h1>
```

```
</div>
```

```
<nav>
```

```
    <ul id="menu-link">
```

```
        <li>
```

```
<a href="index.php">Mājas</a>
</li>
<li>
  <a href="news.php">Jaunumi</a>
</li>
<li>
  <a href="gallery.php">Galerija</a>
</li>
<li>
  <a href="contactus.php">Kontakti</a>
</li>
<li>
  <a href="reservation.php">Rezervācija</a>
</li>
<?php if(isset($_SESSION['user_name'])): ?>
  <?php if($_SESSION['user_type'] == 'admin'): ?>
    <li><a href="admin_page.php">Admin</a></li>
  <?php else: ?>
    <li><a href="profile.php">Profile</a></li>
  <?php endif; ?>
  <li><a href="logout.php">Logout</a></li>
<?php else: ?>
  <li><a href="login_form.php">Login</a></li>
<?php endif; ?>
</ul>
</nav>
</header>

<div class="container">
  <div class="sidebar">
```

<h3>Admin Panelis</h3>

Lietotāji

Aktivitātes

Rezervācijas

Galerija

Ziņojumi

Jaunami

Izrakstīties

</div>

<div class="content">

<div class="users-container">

<h2>Lietotāju Saraksts</h2>

<table class="users-table">

<tr>

<th>Vārds</th>

<th>Uzvārds</th>

<th>E-pasts</th>

<th>Telefona numurs</th>

<th>Lietotāja tips</th>

<th>Darbības</th>

</tr>

<?php

if (mysqli_num_rows(\$users_result) > 0) {

while (\$row = mysqli_fetch_assoc(\$users_result)) {

echo '<tr>';

```

        echo '<td>'.$row['name'].'</td>';
        echo '<td>'.$row['surname'].'</td>';
        echo '<td>'.$row['email'].'</td>';
        echo '<td>'.$row['phone'].'</td>';
        echo '<td>'.$row['user_type'].'</td>';
        echo '<td>
                <a href="#" class="edit-user-btn" data-id="'.$row['id'].'" data-
name="'.$row['name'].'" data-surname="'.$row['surname'].'" data-email="'.$row['email'].'"
data-phone="'.$row['phone'].'" data-user_type="'.$row['user_type'].'">Rediġēt</a>
                <a href="#" class="delete-user-btn" data-id="'.$row['id'].'">Dzēst</a>
        </td>';
        echo '</tr>';
    }
} else {
    echo '<tr><td colspan="7">Nav pievienoti lietotāji.</td></tr>';
}
?>
</table>
</div>

```

```

<div class="activities-container" style='display: none;'>
    <h2>Pievienot Jaunu Aktivitāti</h2>
    <form action="" method="post">
        <h3>Aizpildiet formu</h3>
        <?php
            if(isset($error)){
                echo '<span class="error-msg">'.$error.'</span>';
            }
            if(isset($message)){
                echo '<span class="success-msg">'.$message.'</span>';
            }
        }
    </form>

```

```

    }
    ?>
    <select class="fill-out" name="day" required>
        <option value="" disabled selected>Izvēlieties dienu</option>
        <?php
            foreach ($days as $day) {
                echo "<option value='$day'>$day</option>";
            }
        ?>
    </select>

    <input class="fill-out" type="text" name="activity" required
placeholder="Ievadiet aktivitāti">

    <input class="fill-out" type="text" name="time" required placeholder="Ievadiet
laiku" pattern="\d{2}:\d{2}-\d{2}:\d{2}">

    <input class="fill-out" type="number" name="max_slots" required
placeholder="Ievadiet maksimālo vietu skaitu">

    <input class="form-btn" type="submit" name="add_activity" value="Pievienot">
</form>

<table class="activities-table">
    <tr>
        <th>ID</th>
        <th>Diena</th>
        <th>Aktivitāte</th>
        <th>Laiks</th>
        <th>Maksimālais vietu skaits</th>
        <th>Pieejamais vietu skaits</th>
        <th>Darbības</th>
    </tr>
    <?php
        if (mysqli_num_rows($activities_result) > 0) {

```

```

while ($row = mysqli_fetch_assoc($activities_result)) {
    echo '<tr>';
    echo '<td>'.$row['id'].'</td>';
    echo '<td>'.$row['day'].'</td>';
    echo '<td>'.$row['activity'].'</td>';
    echo '<td>'.$row['time'].'</td>';
    echo '<td>'.$row['max_slots'].'</td>';
    echo '<td>'.$row['available_slots'].'</td>';
    echo '<td>
        <a href="#" class="edit-btn" data-id="'.$row['id'].'" data-
day="'.$row['day'].'" data-activity="'.$row['activity'].'" data-time="'.$row['time'].'" data-
max_slots="'.$row['max_slots'].'">Rediġēt</a>
        <a href="#" class="delete-btn" data-id="'.$row['id'].'">Dzēst</a>
    </td>';
    echo '</tr>';
}
} else {
    echo '<tr><td colspan="7">Nav pievienotas aktivitātes.</td></tr>';
}
?>
</table>
</div>

```

```

<div class="reservations-container" style='display: none;'>

```

```

<h2>Rezervāciju Saraksts</h2>

```

```

<table class="reservations-table">

```

```

<tr>

```

```

    <th>ID</th>

```

```

    <th>Diena</th>

```

```

    <th>Aktivitāte</th>

```

```

        <th>Vārds</th>

        <th>Uzvārds</th>

        <th>Darbības</th>

    </tr>

    <?php
    if (mysqli_num_rows($reservations_result) > 0) {
        while ($row = mysqli_fetch_assoc($reservations_result)) {
            echo '<tr>';
            echo '<td>'.$row['id'].'</td>';
            echo '<td>'.$row['day'].'</td>';
            echo '<td>'.$row['activity'].'</td>';
            echo '<td>'.$row['name'].'</td>';
            echo '<td>'.$row['surname'].'</td>';
            echo '<td>

                <a href="#" class="delete-reservation-btn" data-
id="'.$row['id'].'">Dzēst</a>

            </td>';
            echo '</tr>';
        }
    } else {
        echo '<tr><td colspan="6">Nav pievienotas rezervācijas.</td></tr>';
    }
    ?>

</table>

</div>

```

```

<div class="gallery-container" style='display: none;'>

```

```

    <h2>Pievienot Bildi Galerijai</h2>

```

```

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

```

```

        <h3>Aizpildiet formu</h3>

```



```

<?php
if(isset($error)){
    echo '<span class="error-msg">'.$error.'</span>';
}
if(isset($message)){
    echo '<span class="success-msg">'.$message.'</span>';
}
?>

<input class="fill-out" type="file" name="image" required>

<input class="form-btn" type="submit" name="upload_image"
value="Pievienot">

</form>

<h2>Galerijas Bilde</h2>

<table class="reservations-table">

    <tr>

        <th>Bilde</th>

        <th>Darbības</th>

    </tr>

    <?php
    if (mysqli_num_rows($gallery_result) > 0) {
        while ($row = mysqli_fetch_assoc($gallery_result)) {
            echo '<tr>';

            echo '<td></td>';

            echo '<td>

                <form action="" method="post">

                    <input type="hidden" name="image_id" value="'. $row['id'].'">

                    <input class="form-btn" type="submit" name="delete_image"
value="Dzēst">

                </form>

```

```

        </td>';
    echo '</tr>';
}
} else {
    echo '<tr><td colspan="3">Nav pievienotas bildes.</td></tr>';
}
?>
</table>
</div>

```

```

<div class="messages-container" style='display: none;'>
    <h2>Ziņojumi no klientiem</h2>
    <table class="reservations-table">
        <tr>
            <th>Vārds</th>
            <th>Uzvārds</th>
            <th>Telefona numurs</th>
            <th>E-pasts</th>
            <th>Tēma</th>
            <th>Ziņa</th>
            <th>Izveidots</th>
            <th>Darbības</th>
        </tr>
        <?php
            $messages_query = "SELECT * FROM contact_messages";
            $messages_result = mysqli_query($conn, $messages_query);
            if (mysqli_num_rows($messages_result) > 0) {
                while ($row = mysqli_fetch_assoc($messages_result)) {
                    echo '<tr>';
                    echo '<td>'.$row['fname'].'</td>';

```

```

        echo '<td>'.$row['sname'].'</td>';
        echo '<td>'.$row['phone'].'</td>';
        echo '<td>'.$row['email'].'</td>';
        echo '<td>'.$row['subject'].'</td>';
        echo '<td>'.$row['message'].'</td>';
        echo '<td>'.$row['created_at'].'</td>';
        echo '<td>

                <form action="" method="post">

                    <input type="hidden" name="message_id" value="'.$row['id'].'">

                    <input class="form-btn" type="submit" name="delete_message"
value="Dzēst">

                </form>

        </td>';
        echo '</tr>';
    }
} else {
    echo '<tr><td colspan="9">Nav pievienotas ziņas.</td></tr>';
}
?>
</table>
</div>

```

```

<div class="news-container" style="display: none;">
    <h2>Rediģēt Jaunumu Saturu</h2>
    <form action="" method="post">
        <h3>Jaunumu Saturs</h3>
        <?php
            if(isset($error)){
                echo '<span class="error-msg">'.$error.'</span>';
            }
        }
    </form>

```

```

        if(isset($message)){
            echo '<span class="success-msg">'.$message.'</span>';
        }
    ?>

    <textarea id="news_content" name="news_content"><?php echo
isset($news['content']) ? $news['content'] : ''; ?></textarea>

    <input class="form-btn" type="submit" name="edit_news" value="Atjaunināt">

</form>

</div>

</div>

```

```

<div class="filter-sidebar">
    <h3>Filtrēt</h3>

    <form action="" method="post" id="filter-form">

        <!-- Filters for Activities -->

        <div id="activity-filters">

            <select class="fill-out" name="filter_day">

                <option value="">Visas dienas</option>

                <?php
                foreach ($days as $day) {
                    $selected = ($filter_day == $day) ? 'selected' : '';
                    echo "<option value='$day' $selected>$day</option>";
                }
            ?>

            </select>

            <input class="fill-out" type="text" name="filter_activity" placeholder="Filtrēt
pēc aktivitātes" value="<?php echo $filter_activity; ?>">

        </div>

        <!-- Filters for Users -->

```

```

<div id="user-filters" style="display: none;">

    <input class="fill-out" type="text" name="filter_name" placeholder="Filtrēt pēc vārda" value="<?php echo $filter_name; ?>">

    <input class="fill-out" type="text" name="filter_email" placeholder="Filtrēt pēc e-pasta" value="<?php echo $filter_email; ?>">

    <select class="fill-out" name="filter_user_type">

        <option value="">Visi lietotāju tipi</option>

        <option value="user" <?php echo ($filter_user_type == 'user') ? 'selected' : ''; ?>>Lietotājs</option>

        <option value="admin" <?php echo ($filter_user_type == 'admin') ? 'selected' : ''; ?>>Administrators</option>

    </select>

</div>

```

```

<!-- Filters for Reservations -->

<div id="reservation-filters" style="display: none;">

    <select class="fill-out" name="filter_reservation_day">

        <option value="">Visas dienas</option>

        <?php
        foreach ($days as $day) {

            $selected = ($filter_reservation_day == $day) ? 'selected' : '';

            echo "<option value='$day' $selected>$day</option>";

        }

        ?>

    </select>

    <input class="fill-out" type="text" name="filter_reservation_name"
placeholder="Filtrēt pēc vārda" value="<?php echo $filter_reservation_name; ?>">

    <input class="fill-out" type="text" name="filter_reservation_surname"
placeholder="Filtrēt pēc uzvārda" value="<?php echo $filter_reservation_surname; ?>">

    <input class="fill-out" type="text" name="filter_reservation_activity"
placeholder="Filtrēt pēc aktivitātes" value="<?php echo $filter_reservation_activity; ?>">

</div>

```

```

        <input class="form-btn" type="submit" name="filter" value="Filtrēt">

    </form>

</div>

</div>

<!-- Modal for Editing Activity -->
<div id="editModal" class="modal">
    <div class="modal-content">
        <span class="close">&times;</span>
        <form action="" method="post" id="edit-form">
            <h3>Rediģēt Aktivitāti</h3>
            <input type="hidden" name="id" id="edit-id">
            <select class="fill-out" name="day" id="edit-day" required>
                <?php
                    foreach ($days as $day) {
                        echo "<option value='$day'>$day</option>";
                    }
                ?>
            </select>
            <input class="fill-out" type="text" name="activity" id="edit-activity" required
placeholder="Ievadiet aktivitāti">
            <input class="fill-out" type="text" name="time" id="edit-time" required
placeholder="Ievadiet laiku" pattern="\d{2}:\d{2}-\d{2}:\d{2}">
            <input class="fill-out" type="number" name="max_slots" id="edit-max_slots"
required placeholder="Ievadiet maksimālo vietu skaitu">
            <input class="form-btn" type="submit" name="edit_activity" value="Atjaunināt">
        </form>
    </div>
</div>

<!-- Modal for Editing User -->

```

```
<div id="editUserModal" class="modal">

  <div class="modal-content">

    <span class="close">&times;</span>

    <form action="" method="post" id="edit-user-form">

      <h3>Rediģēt Lietotāju</h3>

      <input type="hidden" name="id" id="edit-user-id">

      <input class="fill-out" type="text" name="name" id="edit-user-name" required
placeholder="Ievadiet vārdu">

      <input class="fill-out" type="text" name="surname" id="edit-user-surname"
required placeholder="Ievadiet uzvārdu">

      <input class="fill-out" type="email" name="email" id="edit-user-email" required
placeholder="Ievadiet e-pastu">

      <input class="fill-out" type="text" name="phone" id="edit-user-phone" required
placeholder="Ievadiet telefona numuru">

      <select class="fill-out" name="user_type" id="edit-user-type" required>

        <option value="user">Lietotājs</option>

        <option value="admin">Administrators</option>

      </select>

      <input class="form-btn" type="submit" name="edit_user" value="Atjaunināt">

    </form>

  </div>

</div>

<!--Footer-->

<footer class="social-media">

  <h4>Ventpils iela 50 k-4, Latvija</h4>

  <h4>+371 (124) 445 88</h4>

  <h4>Pirmdien - Sestdien - 07:00 - 22:00</h4>

  <br>

  <ul>

    <li>
```

<i class="fa-brands fa-square-facebook"></i>

<i class="fa-brands fa-square-instagram"></i>

<i class="fa-brands fa-square-twitter"></i>

</footer>

<!-- font awesome script-->

<script src="https://kit.fontawesome.com/c2f4ffc429.js" crossorigin="anonymous"></script>

<!-- jQuery script -->

<script src="https://code.jquery.com/jquery-3.6.0.min.js" crossorigin="anonymous"></script>

<script>

\$(document).ready(function() {

function showMessage(message, type) {

var msgDiv = \$('<div class="" + (type === 'success' ? 'success-msg' : 'error-msg') + "">' + message + '</div>');

\$('#body').prepend(msgDiv);

setTimeout(function() {

msgDiv.fadeOut('slow', function() {

\$(this).remove();

});


```
    }, 3000);  
}  
  
// Open edit modal  
$('.edit-btn').on('click', function() {  
    $('#edit-id').val($(this).data('id'));  
    $('#edit-day').val($(this).data('day'));  
    $('#edit-activity').val($(this).data('activity'));  
    $('#edit-time').val($(this).data('time'));  
    $('#edit-max_slots').val($(this).data('max_slots'));  
    $('#editModal').show();  
});
```

```
// Open edit user modal  
$('.edit-user-btn').on('click', function() {  
    $('#edit-user-id').val($(this).data('id'));  
    $('#edit-user-name').val($(this).data('name'));  
    $('#edit-user-surname').val($(this).data('surname'));  
    $('#edit-user-phone').val($(this).data('phone'));  
    $('#edit-user-type').val($(this).data('user_type'));  
    $('#editUserModal').show();  
});
```

```
// Close modal  
$('.close').on('click', function() {  
    $('#editModal').hide();  
    $('#editUserModal').hide();  
});
```

```
// Handle delete activity
```

```
$('#delete-btn').on('click', function() {  
    var id = $(this).data('id');  
    $.ajax({  
        url: 'delete_activity.php',  
        type: 'POST',  
        data: { id: id },  
        dataType: 'json',  
        success: function(response) {  
            showMessage(response.message, response.status);  
            if (response.status === 'success') {  
                loadActivities();  
            }  
        }  
    });  
});
```

// Handle delete user

```
$('#delete-user-btn').on('click', function() {  
    var id = $(this).data('id');  
    $.ajax({  
        url: 'delete_user.php',  
        type: 'POST',  
        data: { id: id },  
        dataType: 'json',  
        success: function(response) {  
            showMessage(response.message, response.status);  
            if (response.status === 'success') {  
                loadUsers();  
            }  
        }  
    })  
}
```

```

    });

});

// Handle delete reservation

$('#delete-reservation-btn').on('click', function() {
    var id = $(this).data('id');
    $.ajax({
        url: 'delete_reservation.php',
        type: 'POST',
        data: { id: id },
        dataType: 'json',
        success: function(response) {
            showMessage(response.message, response.status);
            if (response.status === 'success') {
                loadReservations();
            }
        }
    });
});
});

```

```

// Handle edit activity

$('#edit-form').on('submit', function(e) {
    e.preventDefault();
    $.ajax({
        url: 'update_activity.php',
        type: 'POST',
        data: $(this).serialize(),
        dataType: 'json',
        success: function(response) {
            showMessage(response.message, response.status);

```

```

        if (response.status === 'success') {
            $('#editModal').hide();
            loadActivities();
        }
    }
});
});

```

// Handle edit user

```

$('#edit-user-form').on('submit', function(e) {
    e.preventDefault();
    $.ajax({
        url: 'update_user.php',
        type: 'POST',
        data: $(this).serialize(),
        dataType: 'json',
        success: function(response) {
            showMessage(response.message, response.status);
            if (response.status === 'success') {
                $('#editUserModal').hide();
                loadUsers();
            }
        }
    });
});
});

```

// Navigation functionality

```

$('.nav-link').on('click', function(e) {
    e.preventDefault();
    var target = $(this).data('target');

```

```
$('.content > div').hide();

$('.' + target).show();


// Show the correct filters
if (target === 'users-container') {
    $('#user-filters').show();
    $('#activity-filters').hide();
    $('#reservation-filters').hide();
} else if (target === 'activities-container') {
    $('#user-filters').hide();
    $('#activity-filters').show();
    $('#reservation-filters').hide();
} else if (target === 'reservations-container') {
    $('#user-filters').hide();
    $('#activity-filters').hide();
    $('#reservation-filters').show();
} else if (target === 'gallery-container') {
    $('#user-filters').hide();
    $('#activity-filters').hide();
    $('#reservation-filters').hide();
} else if (target === 'news-container') {
    $('#user-filters').hide();
    $('#activity-filters').hide();
    $('#reservation-filters').hide();
}

});


// Show users by default
$('.users-container').show();
$('#user-filters').show();
```

```
$('#activity-filters').hide();  
$('#reservation-filters').hide();  
});  
</script>  
</body>  
  
</html>
```