

Vidzemes Tehnoloģiju un Dizaina tehnikums

Profesionālās vidējās izglītības programma – Programmēšana

Kvalifikācija – Programmēšanas tehniķis

Izglītības ieguves forma – Klātiene

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Tīmekļa lapa "Fantasy Fighting Championship(FFC)"

Autors: IP19 Grupas audzēknis
Ralfs Dambītis

(paraksts)

Cēsis, 2023

ANOTĀCIJA

Tīmekļa lapa "Fantasy Fighting Championship (FFC)". Dambītis, Ralfs.
Kvalifikācijas darbs, 67 lappuses, 69 attēli, 9 tabulas, 23 literatūras avoti, 7 pielikumi.

Kvalifikācijas darba mērķis ir izstrādāt jautru un konkurētspējīgu tīmekļa vietni priekš draugiem un jauktās cīņas mākslas (angļu v. *Mixed Martial Arts*, saīsinājumā *MMA*) faniem, kas rosina sāncensību starp vienu un otru paralēli padarot to jautru. Darbā ir aprakstīts, kādas prasības tika izvirzītas, lai sasniegtu rezultātu, kā arī tiek attēlots, kuras no prasībām ir īstenotas. Dažas no svarīgākajām nodaļām ir literatūras apskats un variantu salīdzinājums, kur tiek salīdzinātas citas vietnes ar izstrādāto vietni, prasību specifikācijā tiek rakstīts par nepieciešamajām prasībām, lai izstrādātu vietni. Kā arī ir izstrādāts funkcionālais un dinamiskais sistēmu modelis, kas attēlo lietotāja kustību caur vietni un datus, kuri tiek pārnesti. Lietotāja ceļvedis palīdz lietotājam paskaidrot un parādīt, kā tīmekļa vietne darbojas un kā notiek kustība pār vietni.

Darba mērķis tika sasniegts un tīmekļa vietnes tika izstrādāta, balstoties uz izvirzīto mērķi. Vietne ir konkurētspējīga un ļauj lietotājiem savstarpēji sacensties, kas padara to par jautru pieredzi jebkuram. Veikt likmes uz cīņām un pēc tam iegūt to rezultātu ir iespējams jebkuram lietotājam, kas piedalās tīmekļa vietnē.

ANOTATION

The web page "Fantasy Fighting Championship (FFC)". Dambītis, Ralfs. Qualification paper, 67 pages, 69 images, 9 tables, 23 literature sources, 7 appendices.

The aim of the qualification paper is to develop a fun and competitive website for friends and fans of *Mixed Martial Arts* (MMA) that encourages rivalry between participants while making it enjoyable. The paper describes the requirements set to achieve the desired outcome and illustrates which of these requirements have been implemented. Some of the most important aspects include a literature review and a comparison of other websites with the developed site. The specification of requirements outlines the necessary elements for website development. Additionally, functional and dynamic system models have been created to depict user navigation and the data transferred by the website. A user guide assists in explaining and demonstrating how the website functions and how users can navigate it.

The aim of the paper has been achieved, and the website has been developed based on the set objectives. The website is competitive and allows users to compete with each other, making it an enjoyable experience for everyone. Users can place bets on fights and subsequently obtain the results, engaging all participants on the website.

SATURA RĀDĪTĀJS

Anotācija.....	2
Anotation	3
Satura rādītājs	4
Ievads.....	5
1. Literatūras apskats un variantu salīdzinājums	6
2. Prasību specifikācija	10
2.1. Funkcionālās prasības.....	10
2.2. Nefunkcionālās prasības	13
3. Mērķa sasniegšanas līdzekļu izvēle un pamatojums	15
4. Sistēmas struktūras modelis.....	17
5. Funkcionālais un dinamiskais sistēmas modelis	24
6. Datu struktūras apraksts.....	26
7. Reaģējoša tīmekļa vietne	28
7.1. Mazie ekrāni	28
7.2. Vidējie ekrāni	31
7.3. Lielie ekrāni	33
8. Testēšana	36
9. Lietotāja ceļvedis	52
Secinājumi un priekšlikumi	63
Izmantotā literatūra	65
Pielikumi.....	67
1. pielikums. Lietotāja reģistrācijas programmas kods.....	68
2. pielikums. Lietotāja pieslēgšanās programmas kods	71
3. pielikums. Skata mainīšanas programmas kods.....	73
4. pielikums. Lietotāja sesijas programmas kods.....	75
5. pielikums. Lietotāja dzēšanas no datu bāzes porgrammas kods	76
6. pielikums. Likmju ievietošanas progrmmas kods.....	78
7. pielikums. Likmju rezultātiem porgrammas kods.....	83

IEVADS

Cīņas sports ir populārs sporta veids, kuram seko līdzī sākot no 500 miljonu līdz 2,5 miljardu skatītāju, no kuriem 75 % – 80 % ir vīrieši. Šis sporta veids ir zināms vislabāk vecuma grupās sākot no 16 gadiem līdz 44 gadiem, kura sastāvs ir 70 % no visu skatītāju skaita (Bloom, 2022; Marconi 2022).

Ultimate Fighting Championship, saīsinājumā *UFC*) ir vispopulārākā jauktās cīņas mākslas (angļu v. *Mixed Martial Arts*, saīsinājumā *MMA*) kompānija, kur cīkstoņi cīnās, lai kļūtu par pasaules čempioniem. Cilvēkiem ir izdevība likt likmes uz cīkstoni, kurš uzvarēs pa īstu naudu, kas var izraisīt atkarību cilvēkā, bet ir risinājums un to sauc Fantāzijas Cīņu Čempionāts (angļu v. *Fantasy Fighting Championship*, saīsinājumā, saīsinājumā *FFC*) tur var izmantot virtuālo, iekš spēles naudu, kur spēlētājs netērēs savu naudu, bet spēles naudu tas nozīmē, ka cilvēki ar tendenci izveidot atkarību varēs šeit izpausties, netērējot savus ienākumus paralēli sacenšoties ar citiem spēlētājiem.

Mērķis

Kvalifikācijas darba mērķis ir izstrādāt jauktu un konkurētspējīgu tīmekļa vietni priekš draugiem un jauktās cīņas mākslas (angļu v. *Mixed Martial Arts*, saīsinājumā *MMA*) faniem, kas rosina sāncensību starp vienu un otru paralēli padarot to jauktu.

Uzdevumi

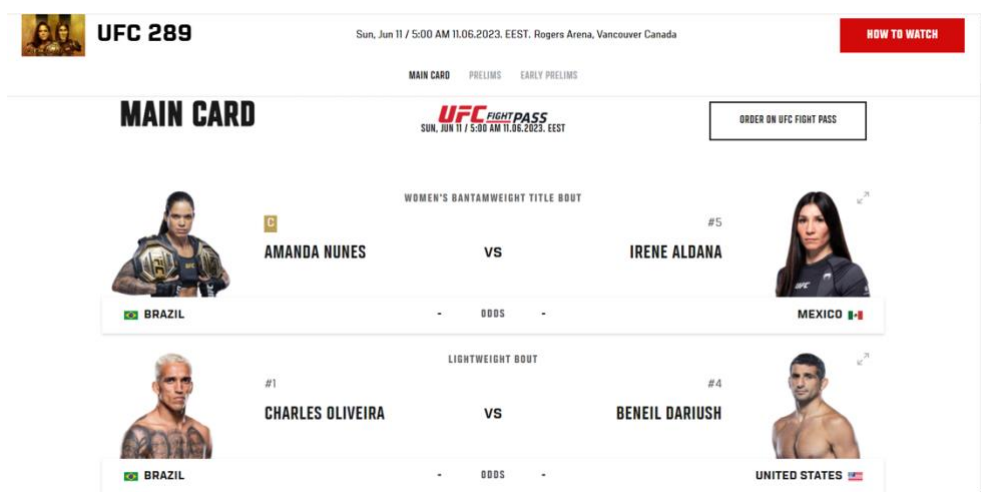
1. Izveidot tīmekļa vietnes dizainu.
2. Izstrādāt priekšgalsistēmu (angļu v. *front-end*), kurā lietotāji, izmantojot virtuālo spēles naudu, varēs likt likmes uz cīņām.
3. Izveidot datu bāzes arhitektūru ER diagrammu veidā, kā arī izstrādāt pašu datu bāzi.
4. Izveidot aizmugursistēmu (angļu v. *back-end*), kura saglabās datubāzē lietotāju veikto likmju informāciju un vispārēju informāciju par lietotāju.
5. Izstrādāt testus un veikt tīmekļa vietnes testēšanu.
6. Novērst radušās problēmas un veikt labojumus.

1. LITERATŪRAS APSKATS UN VARIANTU SALĪDZINĀJUMS

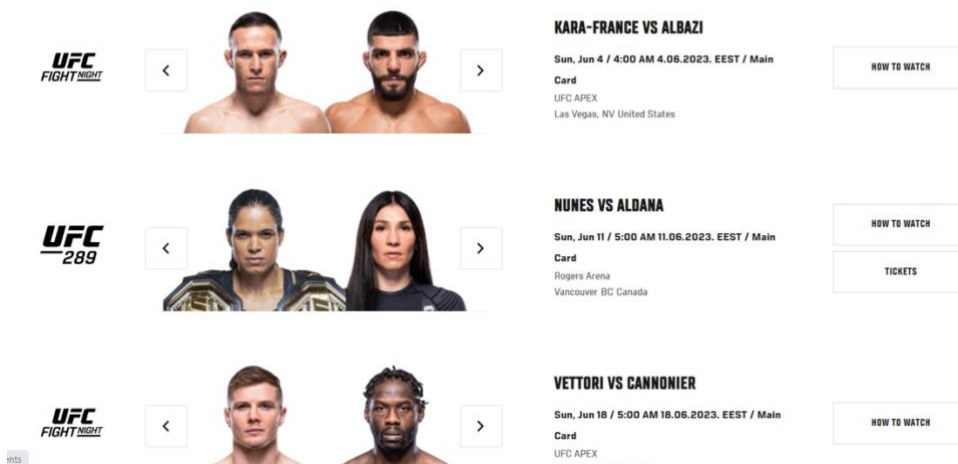
UFC ir jauktās cīņas mākslas (angļu v. *mixed martial arts*) reklamēšanas kompānija, kas ļauj cīkstoņiem savstarpēji konkurēt un kāpt augstāk pa ranga kāpnēm uz čempiona titulu.

Izstrādātajam darbam līdzvērtīgi projekti jau eksistē, un tie tiek aprakstīti šajā nodaļā. Pirmais variants ir <https://www.ufc.com/>, kas ir UFC oficiālā mājaslapa. Otrais variants ir <https://www.oddsshark.com/ufc>, kas nav tikai fokusēts uz MMA sporta veidu, bet tā arī ir tīmekļu vietne ar UFC datiem.

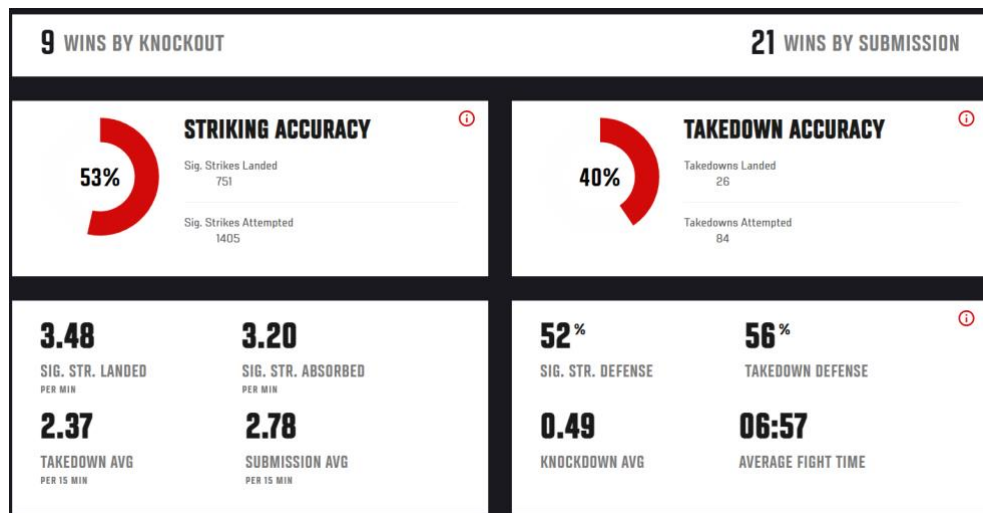
Šīm vietnēm ir kopīgs tas, ka abas vietnes attēlo UFC cīņu un cīkstoņu statistiku (skatīt 3. attēlu, 7. attēlu), cīkstoņu rangu (skatīt 5. attēlu, 9. attēlu.) kā arī tuvākās cīņas un to cīkstoņu koeficientus (skatīt 1. attēlu, 2. attēlu, 6. attēlu.), kas norāda cīņas favorītu. Šīs vietnes arī informē lietotājus par jaunākajiem notikumiem UFC pasaulē, jaunumu sadaļā (skatīt 4. attēlu, 8. attēlu.).



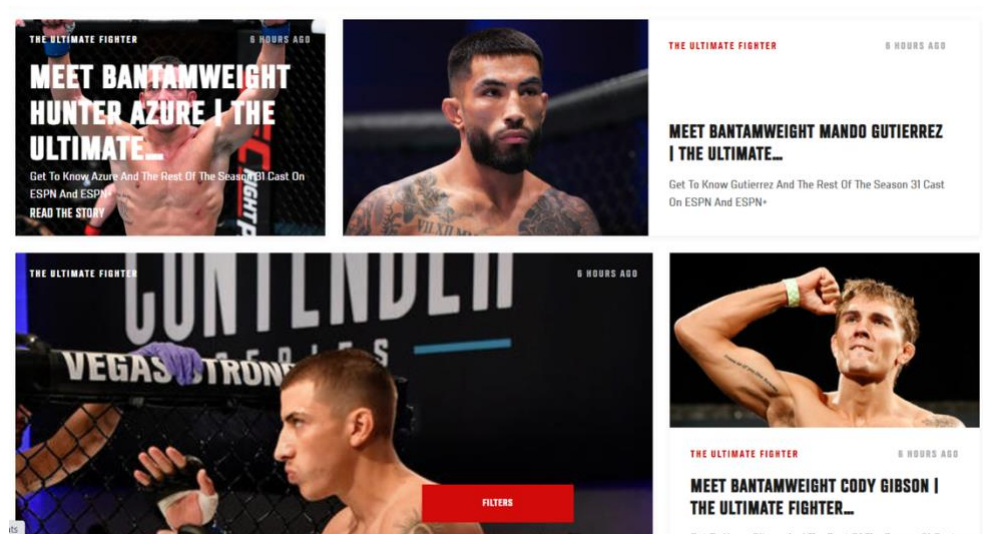
1. attēls. UFC lapas notikumi un tā dalībnieki



2. attēls. UFC lapas notikumi



3. attēls. UFC lapas cīkstoņa statistika



4. attēls. UFC lapas jaunumi

ATHLETE RANKINGS

MEN'S POUND-FOUR-POUND TOP RANK	FLYWEIGHT	BANTAMWEIGHT	FEATHERWEIGHT
JON JONES	BRANDON MORENO	ALJAMAIN STERLING	ALEXANDER VOLKANOVSKI
CHAMPION	CHAMPION	CHAMPION	CHAMPION
1 Jon Jones	1 Delveson Figueiredo	1 Merab Dvalishvili	1 Yair Rodriguez
2 Alexander Volkanovski	2 Alexandre Pantoja	2 Sean O'Malley	2 Max Holloway
3 Islam Makhachev	3 Kai Kara-France	3 Henry Cejudo	3 Brian Ortega
4 Leon Edwards	4 Brandon Royval	4 Cory Sandhagen	4 Arnold Allen
5 Israel Adesanya	5 Mathheus Nicolau	5 Petr Yan	5 Josh Emmett
6 Aljamain Sterling	6 Alex Perez	6 Marlon Vera	6 Calvin Kattar

5. attēls. UFC lapas cīkstoņu rangs

Saturday, June 3

- UFC Fight Night : Kara-...

Opening

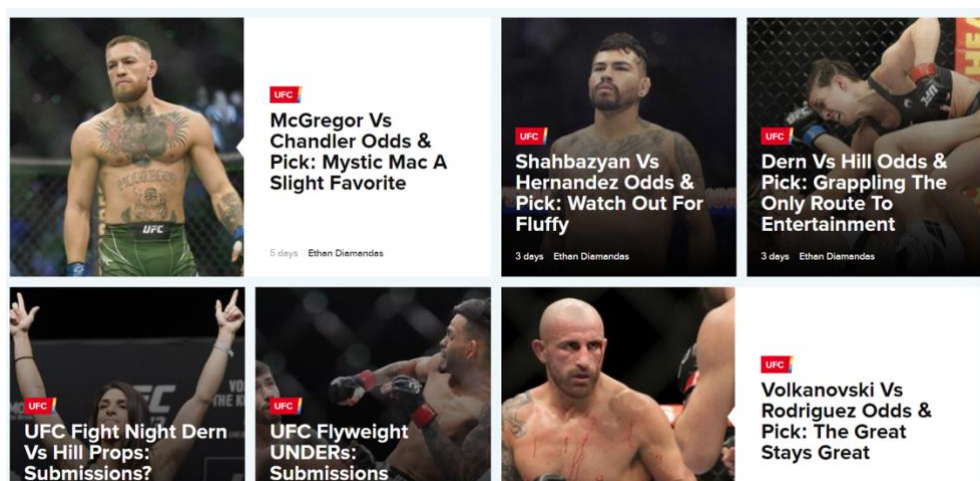
Current

9:00pm ET	Kai Kara-Fra... vs Amir Albazi	Moneyline	Moneyline
		-119	+110
<div>Matchup</div>		-111	-130
9:00pm ET	Alex Caceres vs Daniel Pineda	Moneyline	Moneyline
		-175	-173
<div>Matchup</div>		+150	+148
9:00pm ET	Jared Gordon vs Jim Miller	Moneyline	Moneyline
		-275	-181
		+235	+156
9:00pm ET	Tim Elliott vs Victor Altam...	Moneyline	Moneyline
		-111	-160
<div>Matchup</div>		-119	+140

6. attēls. Oddsshark lapas notikumi un tā dalībnieki

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Last, First						W	L	Age	HT	WT	Stance	Reach							
Abdelwahab, Hamdy								30	6'2"	260lbs	N/A	72"							
Abdurakhimov, Shamil						20	4	41	6'3"	235lbs	Orthodox	75"							
Abreu, Klidson						15	3	30	6'2"	235lbs	Orthodox	73"							
Adashev, Zarrukh								30	5'6"	134lbs	N/A	64"							
Adesanya, Israel						18	0	33	6'3"	185lbs	Switch	79"							
Agepova, Mariya						6	0	26	5'6"	123lbs	N/A	N/A							
Aguilar, Kevin						17	2	34	5'6"	145lbs	Orthodox	72"							
Aguilera, Christian								31	5'8"	169lbs	N/A	N/A							
Aguirre, Nick								27	N/A	N/A	N/A	N/A							
Akhmedov, Omari						19	4	35	6'0"	169lbs	Orthodox	72"							
Albazi, Amir								29	5'6"	125lbs	N/A	68"							
Aldana, Irene						10	5	35	5'8"	134lbs	Orthodox	67"							
Aldo, Jose						28	6	36	5'6"	145lbs	Orthodox	69"							
Aldrich, Jj						8	3	30	5'4"	114lbs	Southpaw	66"							
Alekseeva, Irina								32	5'7"	130lbs	N/A	25"							
Alexander, Lucas								27	5'10"	145lbs	N/A	N/A							

7. attēls. Oddsshark lapas cīkstoņu statistika



8. attēls. Oddsshark lapas jaunumi

All ▼

Lightweight			
Rank	Fighter	Record	Last 3
★	Khabib Nurmagomedov	28-0-0	W-W-W
1	Dustin Poirier	25-5-0 (1 NC)	W-W-W
2	Tony Ferguson	26-3-0	W-W-W
3	Conor McGregor	21-4-0	L-W-W
4	Donald Cerrone	36-14-0 (1 NC)	L-W-W
5	Justin Gaethje	21-2-0	W-W-L
6	Al Iaquinta	14-4-1	L-W-L
7	Edson Barboza	20-8-0	L-W-L
8	Kevin Lee	18-5-0	L-L-W
9	Paul Felder	17-4-0	W-L-W
10	Charles Oliveira	28-8-0 (1 NC)	W-W-W
11	Gregor Gillespie	13-0-0	W-W-W
12	Alexander Hernandez	11-2-0	L-W-W

9. attēls. Oddsshark lapas cīkstoņu rangs

Mana vietne atšķiras no šīm vietnēm, ka lietotāji sacenšas savstarpēji un viņi liek likmes ar virtuālo naudu, kas samazina atkarības iespēju. Šī funkcija nav nevienā no šīm vietnēm. Kaut gan manā vietnē nav jaunākā informācija par notiekošo UFC pasaulē un tik detalizēta informācija par statistiku, mana vietne atšķiras ar to, ka man lietotājiem tiek dota iespēja draudzīgi savstarpēji konkurēt vienam ar otru un pierādīt to, ka spēj labāk uzzināt cīņu iznākumu nekā citi lietotāji. Kā arī manas vietnes dizains un izskats ir lietotājiem draudzīgāks un pievilcīgāks.

2. PRASĪBU SPECIFIKĀCIJA

Prasību specifikācija ir būtiska, lai izstrādātu projektu un zinātu kādai ir jābūt tā funkcionalitātei (Presentationeze, 2013). Veicina sapratni par projekta mērķi un plānu, kas palīdz sasniegt gala produkta izstrādi. Atvieglo izstrādi un testēšanu, definē projekta apjomu kā arī palīdz novērst kļūdas. Bez skaidras un akurātas prasību specifikācijas ir risks saskarties ar neveiksmīgi izstrādātu produktu, novirzīties no mērķa un nesaprašanās.

2.1. FUNKCIONĀLĀS PRASĪBAS

Funkcionālās prasības ir prasības, kuras izstrādātājam ir jāievieš, lai lietotāji varētu veikt savas darbības (Altexsoft, 2022). Tāpēc ir svarīgi tās norādīt, lai jebkurš, tās lasot, spētu saprast galveno domu un funkciju. Pārsvārā funkcionālās prasības tiek noteiktas ar specifiskiem raksturojumiem un nosacījumiem (skatīt 1. tabulā).

1. tabula

Funkcionālās prasības

Nr.	Nosaukums	Apraksts
1.	Tīmekļa vietnes augšējā navigācijas josla vai galvene.	Uzejot virsū kādam no virsrakstiem ar kursoru, tiek pasvītroti ar apakš svītru zem teksta. Uzspiežot virsū uz kādam no navigācijas sadaļām, lietotāju pārvieto uz attiecīgo lapu.
2.	Tīmekļa vietnes apakšējā navigācijas josla vai kājene.	Tīmekļa vietnes apakšā atradīsies sociālo mediju ikonas un tālāka informācija par vietni. Ar kursoru uzbīdot uz sociālo mediju ikonām, tās maina krāsu. Uzspiežot uz attiecīgo ikonu, lietotāju pārvietos uz sociālo mediju vietnēm.
3.	Iziet (angļu v. <i>EXIT</i>) poga	<i>EXIT</i> poga, kas lietotājam ļaus iziet no sava profila. Uzbīdot uz pogas, tā maina krāsu un tiek pasvītrotā zem teksta. Uzspiežot uz pogas lietotāju pārvieto uz pieslēgšanās (angļu v. <i>Login</i>), reģistrācijas (angļu v. <i>Register</i>) tīmekļa vietne.
4.	Lietotāja vārds	Uz lielajiem ekrāniem pieslēdzoties lapai, kad lietotājs ir pieslēdzies, viņa lietotājvārds parādās pie vietnes augšējās navigācijas joslas.
5.	Aizmugures fons(angļu v. <i>background</i>) galvenajai lapai tīmekļa vietnei	Galvenās lapas tīmekļa vietnē fons tiek piemērots ar nākošo lielo pasākumu <i>Ultimate fighting championship (UFC)</i> .
6.	Pieteikšanās (angļu v. <i>login</i>)	Lietotājam nepieciešams pieteikties tīmekļa vietnei, lai tā datus saglabātu, un lietotājs spētu likt jebkāda veida likmes. Pie vietnes pieslēgties (angļu v. <i>Login</i>) klāt ir e-pasta adrese un parole. Lai pabeigtu pieteikšanos un virzītos uz vietnes sāukmlapu, jāspiež poga <i>pieslēgties</i> (angļu v. <i>Login</i>).

		Ja lietotājs nav pieteicies ir iespēja uzspiest blakus pogu reģistrēties (angļu v. <i>Register</i>).
7.	Reģistrācija (angļu v. <i>register</i>)	Jauniem lietotājiem nepieciešams reģistrēties, lai spētu veikt izmaiņas un nokļūtu galvenajā lapā. Reģistrācijai būs nepieciešams aizpildīt 4 datu ievadlaukus: lietotājvārds (angļu v. <i>nickname</i>), e-pasta adrese (angļu v. <i>e-mail</i>), parole (angļu v. <i>password</i>) un atkārtotā parole (angļu v. <i>repeat password</i>). Ja lietotājam jau ir konts blakus reģistrācijai ir pieslēgties (angļu v. <i>Login</i>) poga.
8.	Bez konta pieslēgšanās	Lietotājs bez izveidota konta, nevar pieslēgties sākulapā vietnei un veikt jeb kāda veida darbības tajā.
9.	Ikona profils	Blakus lietotājvārdam uz navigācijas joslas ir ikona, kas attēlo lietotāju. Uzspiežot uz ikonas tā iekrāsojas un lietotāju aizved uz viņa lietotāja iestatījumiem. Uz mazākiem ekrāniem šī ikona atrodas pie burger navigācijas
10.	Konta sadaļa	Tīmekļa vietnes konta sadaļā lietotājam ir vairāki iestatījumi. Lietotājs var mainīt paroli un e-pasta adresi. Ievadīt vārdu, uzvārdu, vecumu un lokāciju kā arī pievienot attēlu. Ja lietotājs vēlas dzēst kontu, tad šo var veikt šajā sadaļā.
11.	Logo	Logo novieto pie tīmekļa vietnes navigācijas. Uzspiežot uz logo lietotāju novieto atpakaļ pie sākulapā sākumu.
12.	Hipersaite <i>aplūkot rangus</i> (angļu v. <i>Check out the rankings</i>).	Nospiežot uz hipersaites <i>aplūkot rangus</i> (angļu v. <i>Check out the rankings</i>), lietotāju pārvieto uz tīmekļa vietni rangi (angļu v. <i>ranks</i>).
13.	Hipersaite <i>Kas ir FFC?</i> (angļu v. <i>What is FFC?</i>).	Nospiežot uz hipersaites <i>Kas ir FFC?</i> (angļu v. <i>What is FFC?</i>), lietotāju pārvieto uz tīmekļa vietni <i>informācija</i> (angļu v. <i>Info</i>).
14.	Hamburger navigācija	Ekrānu samazinot vai lietojot mazāku par lielo, datora monitora ekrānu, piemēram, telefonu vai planšeti, augšējā navigācija pārvēršas par burgera navigāciju, kas ir 3 paralēlas līnijas vertikāli. Uzspiežot atver vaļā navigāciju, kur parādās visi virsraksti uz kuriem uzspiežot aizmet uz izvēlēto vietni. Pārejot pāri virsrakstam tā lauciņš maina krāsu. Šī navigācija ir mazāka izmēra tīmekļa vietnes navigācijas josla.
15.	Sākulapā FFC apraksts	Zem galvenās lapas fona (angļu v. <i>background</i>) atrodas apraksts par <i>fantāzijas cīņu čempionātu</i> (angļu v. <i>FFC</i>).
16.	Sākums (angļu v. <i>home</i>)	Augšējā navigācijā atrodas virsraksts sākulapa (angļu v. <i>Home</i>), kuru uzspiežot lietotāju atgriež uz sākuma lapu.

		Uz mazākiem ekrāniem šis virsraksts atrodas pie burger navigācijas
17.	Rangi (angļu v. <i>ranks</i>)	Augšējā navigācijā atrodas virsraksts <i>rangi</i> (angļu v. <i>Ranks</i>), kuru uzspiežot lietotāju pārvieto uz tās vietnes sadaļu. Uz mazākiem ekrāniem šis virsraksts atrodas pie burger navigācijas
18.	Likmes (angļu v. <i>bets</i>)	Augšējā navigācijā atrodas virsraksts <i>likmes</i> (angļu v. <i>bets</i>), kuru uzspiežot lietotāju pārvieto uz tās vietnes sadaļu. Uz mazākiem ekrāniem šis virsraksts atrodas pie burger navigācijas
19.	Informācija(angļu v. <i>info</i>)	Augšējā navigācijā atrodas virsraksts <i>informācija</i> (angļu v. <i>Info</i>), kuru uzspiežot lietotāju pārvieto uz vietnes sadaļu. Uz mazākiem ekrāniem šis virsraksts atrodas pie burger navigācijas
20.	Rezultāti (angļu v. <i>Results</i>)	Augšējā navigācijā atrodas virsraksts <i>rezultāti</i> (angļu v. <i>Results</i>), kuru uzspiežot lietotāju pārvieto uz vietnes sadaļu. Uz mazākiem ekrāniem šis virsraksts atrodas pie burger navigācijas
21.	Likmju daudzums un skaits	Likmes tiks sadalītas vairākās iedaļās sākot no 10 līdz 200. (10, 20, 50, 100, 200).
22.	Izvēles rutiņas	Iezīmējot vienu no rutiņām lietotājs tā atzīmē izvēlēta cīkstoņa datus. Abas pogas nevar būt iezīmētas paralēli.
23.	Uzspiežot uz pogas	Kad lietotājs nospiež pogu, atlikušās pogas nobloķējas un cīņas, un cīkstoņa dati kā arī likmes summa, tiek ievietoti datu bāzē.
24.	Dati par cīņu	Katrai cīņai būs lauciņš, kas attēlos katru cīņas posmu. Kā arī cīkstoņa datus Kopā ir 3 posmi cīņām tie ir - Galvenā kārts(angļu v. <i>Main card</i>), Prēlimi (angļu v. <i>Prelims</i>), Agrie prēlimi (angļu v. <i>Early prelims</i>). Cīņas tiek iedalītas pēc tā vai tā ir titul cīņā (angļu v. <i>title bout</i>) vai parastā cīņa (angļu v. <i>regular bout</i>). Cīņu veidam tiek pievienot klāt svara kategorija, kurā notiek cīņa. Pievienotai informācijai klāt ietilpst - vārds, uzvārds, numerācija, svara kategorijā un koeficients.
25.	Cīņas pasākuma dati	Virs cīkstoņa pievieno datus par pašu pasākumu. Pirmie dati, kas parādās ir kurš pasākums tas ir , piemēram, <i>UFC 283</i> .
26.	Cīņu rezultāti	Sadaļa priekš likmju rezultātiem, attēlojot likmju rezultātus. Šo varēs redzēt navigācijas joslas sadaļā rezultāti (angļu v. <i>Results</i>). Rezultāti tiks izvadīti kastēs vairākās, attēlojot uzvarētāja statistiku.

27.	Informācijas vietnes kastes	Lietotājam tinot zemāk, informācijas vietnes kastes palēnām ieplūst ekrānā un parādās lietotāja redzes lokā.
28.	API	Atrasto API savieno un pārveido ar mājaslapas palīdzību, lai spētu sašķirot datus un sagatavot tos priekš datu bāzes ievades. Atrast noteiktos API tieši priekš noteiktajiem datiem. Izveidot veidu kā šis API tiek izsaukt noteiktos laikos.
29.	Valūtas atjaunināšana	Pēc katras veiksmīgas likmes, lietotājam pievienojoties vietnei, atjaunojas virtuālā nauda, ja veiktās likmes cīkstonis ir uzvarējis.
30.	Virtuālā valūta	Izveidot virtuālo valūtu, kas katram lietotājam būs specifiska un tās daudzums limitēts. Šo valūtu varēs tērēt priekš cīņas likmēm. Katram lietotājam sākumā tik iedota summa 1000, kuru būs iespējams tērēt likmēs.
31.	Datu bāze priekš lietotājiem	Izveidot specifiski datu bāzi, kas saglabās datus priekš lietotājiem. Lietotāju dati būs e-pasts, parole, lietotājvārds, vārds, uzvārds utt. Šajā datu bāzē glabās datus priekš lietotāju iekrātās un dabūtās summas no likmēm vai sākumā iedotās virtuālās naudas.
32.	Datu bāze priekš API	Izveidot specifisku datu bāzi priekš API, kur dati no API tiks ievilkti datu bāzē. Ievilkto dati pēc tam tiek sadalīti un savadīti attiecīgajos lauciņos priekš izvades tīmekļa vietnē.

2.2. NEFUNKCIONĀLĀS PRASĪBAS

Nefunkcionālās ir saistītas ar sistēmas arhitektūru, veikspēju, drošību, uzturamību un citiem aspektiem, kas nav tieši saistīti ar tās funkcionalitāti (Altexsoft, 2022). Tās ir saistītas ar citiem aspektiem, kas atbild par sistēmas darbību un veikspēju (skatīt 2. tabulu).

2. tabula

Nefunkcionālās prasības

Nr.	Nosaukums	Apraksts
33.	Tīmekļa lapas stils	Tīmekļa lapas stils atvieglo redzamību, nekas nav pārāk spožs. Lapas stils balstās pēc krāsu paletes, kas ir izvēlēta un izmantota priekš tīmekļu lapas. Visi lapas elementi ir viegli salasāmi un saredzami.
34.	Informācija	Informācija tīmekļa vietnē ir viegli saprotama un salasāma, lai atvieglotu lietotājam pieredzi ar vietni.
35.	Valoda	Tīmekļa vietnes valoda ir angļu valodā.
36.	Uz lielajiem ekrāniem navigācija	Augšējā navigācija uz lielajiem ekrāniem ir salikta rindā horizontāli.

37.	Uz mazajiem ekrāniem navigācija	Uz mazajiem ekrāniem navigāciju var atvērt uzspiežot uz tā saukto burgeri, kas atvērs visus virsrakstus kolonnā.
38.	Reaģējoša navigācija	Samazinot ekrānu uz mazāku, horizontālā rinda ar virsrakstiem samazinās un pārvēršas par burgera pogu.
39.	Pieslēgšanās un reģistrācijas fons	Pieslēgšanās un reģistrācijas fons ir bilde, kas tiek izmantota, lai attēlotu cīņas sportu
40.	Reaģējoša pieslēgšanās un reģistrācijas	Pieslēgšanās un reģistrācijas kastei, kas ietver visus teksta lauciņus jācentrējas uz vidu.
41.	Reaģējoša pieslēgšanās un reģistrācija uz lielajiem ekrāniem	Fona bildei jābūt redzamai aizmugures teksta lauciņu kastei. Tā atrodas vidū tīmekļa vietnei.
42.	Reaģējoša pieslēgšanās un reģistrācija uz mazākiem ekrāniem	Fona bilde pie mazākiem ekrāniem pazūd. Teksta lauciņu kaste atrodas pa visu tīmekļa vietni tā, lai visus lauciņus varētu viegli saredzēt.
43.	Reaģējoša tīmekļa vietne pēc pieslēgšanās	Pēc pieslēgšanās tīmekļa vietne ir reaģējoša un strādā uz visiem ekrāniem. Tās skaitā ir <i>sākulapa</i> (angļu v. <i>Home</i>), <i>rangi</i> (angļu v. <i>Ranks</i>), <i>likmes</i> (angļu v. <i>Bets</i>), <i>informācija</i> (angļu v. <i>Info</i>), <i>rezultāti</i> (angļu v. <i>Results</i>) un <i>profila</i> (angļu v. <i>Profile</i>) vietnes.
44.	Reaģējoša <i>ranga</i> (angļu v. <i>Ranks</i>) vietne.	Skatoties uz mazākiem ekrāniem 3 labāko lietotāju kolonna sašaurinās platumā. 50 labāko lietotāju tabula samazinās un sašaurinās.
45.	Reaģējoša <i>likmju</i> (angļu v. <i>Bets</i>) vietne.	Skatoties uz mazākiem ekrāniem virsraksts samazinās. Izvēles rūtiņas, cīkstoņu dati un likmju pogas samazinās padarot tos vieglāk saredzamus uz mazākiem ekrāniem.
46.	Reaģējoša <i>informācijas</i> (angļu v. <i>Info</i>) vietne.	Skatoties uz mazākiem ekrāniem informācijas kastes samazinās. Jo mazāks paliek ekrāns, jo tuvāk kastes tuvojas centram.
47.	Reaģējoša <i>rezultātu</i> (angļu v. <i>Results</i>) vietne.	Skatoties uz mazākiem ekrāniem rezultātu kastes tiek centrētas un novietotas, lai tās spētu saredzēt kopā ar to datiem.
48.	Reaģējoša <i>profila</i> (angļu v. <i>Profile</i>) vietne.	Skatoties uz mazākiem ekrāniem ievad lauki un navigācijas sadaļa priekš vietnes samazinās, lai spētu informācija ielipt ekrānā.
49.	Likmju kastes statistika.	Katrai kastei informācija un cīkstoņa statistika ir viegli salasāma. Kastēm dizains ir viegli saprotams un pievilcīgs.
50.	Ranga (angļu v. <i>Ranks</i>) noteikumi.	Zem lietotāja rangiem ir aprakstīti noteikumi par ranga sistēmu un kā tā strādā.

3. MĒRĶA SASNIEGŠANAS LĪDZEKĻU IZVĒLE UN PAMATOJUMS

HTML – Hiperteksta iezīmēšanas valoda (HTML), kas ir paredzēta priekš tīmekļa lappušu un citu pārlūkprogrammu attēlojamās informācijas glabāšanai (O'Grady, 2022). Bez šīs valodas nespētu veidot tīmekļa vietnes, jo tā definē un ievieto elementus, kas satur galveno informāciju par vietni. Šajā vietnē HTML palīdz lietotājiem pārvietoties uz citām lapām kā arī redzēt visu informāciju, kas ir saistīta ar vietni. Kopumā tas saslēdz vietni ar failiem, rindkopām, sarakstiem, saitēm, attēliem un lapas atrašanos vietu.

CSS – Stila lapas kaskadēšana (CSS) ir īpaša stila lapas valoda, tā atbild par tīmekļa vietnes dizainu. CSS piemēro stilu pie HTML elementiem un novieto tos noteiktajās vietās, dod lielumus un pozīcijas (Moris, 2022). Šajā vietnē CSS tiek izmantots diezgan maz, bet to izmanto, lai pievienotu noteiktas stila vērtības, kuras nevar izdarīt ar *Tailwind*.

Javascript – skriptu valoda, kas tiek izmantota, lai radītu interaktīvas un dinamiskas funkcijas tīmekļa lapās (Paruch, 2023). Savā vietnē es izmantošu *Javascript*, lai veidotu funkcijas, kas palīdz lietotājam pārvietoties no lapas uz lapu vai datu pievienošanas bez liekas lapas pārlādēšanas un daudz citu veidu funkciju.

PHP – servera puses skriptu valoda, kas ir izstrādāta tīmekļu vietnes izveidei un dinamiskai darbībai, tā atbalsta formu apstrādi, datu bāzes pieprasījumus un citas funkcijas. Priekš šīs tīmekļu vietnes es PHP izmantoju, lai ievadītu un izvadītu datus no datu bāzēm kā arī veiktu aprēķinus priekš daudzām funkcijām (Kolade, 2021).

Tailwind – CSS rīku kopa, kas palīdz ātrāk un efektīvāk izveidot stilu un izkārtojumu tīmekļu vietnēm. Šis rīku komplekts iekļauj lielu skaitu klases, kas atvieglo izkārtojumu un stilu definēšanu. Tas piedāvā arī saderīgus un pielāgojamus iepriekš definētus dizainus un krāsu paleti, kas palīdz ātri un vienkārši pielāgot vietnes izskatu. *Tailwind* es izmantošu priekš lapas stila, izkārtojuma, izmēriem un pārsvarā visu, kas ir saistīts ar stilu (Fitzgerald, 2022).

MySQL – strukturētajām vaicājumu valodām (*MySQL*), ko lieto, lai manipulētu ar datiem un informāciju, kas glabājas datubāzē (B. Richard, 2023). Tā ļauj izveidot, uzlabot un atgūt informāciju no datubāzes, veikt datu konsolidāciju, salīdzināšanu, ierobežojumus un daudz ko citu.

Cron-job – laika plānots uzdevums, kas tiek izpildīts automātiski. To var izmantot, lai iestatītu atkārtotas darbības noteiktos laikos vai noteiktos intervālos, piemēram, ik dienas, ik mēnesi vai ik stundu (Kubernetes, 2023). *Cron-job* iekšā ir saraksts ar uzdevumiem, kur katram uzdevumam ir norādīts izpildes laiks un komanda, kas jāizpilda. Šie uzdevumi var ietvert datu

dublēšanu, failu sistēmas tīrīšanu, datubāzes kopijas veidošanu vai jebkādu citu uzdevumu, kas jāizpilda automātiski. Izmantots ir produkts, kas pieejams <https://cron-job.org/en/>.

IconScout – pielāgojama digitālo līdzekļu bibliotēka ar vairāk nekā 4,4 miljoniem ikonu, ilustrāciju un *Lottie* animācijām. Šajā darbā *IconScout* tiek izmantots priekš ikonu piesaistīšanas darba vidē. Kā var redzēt, daudzas ikonas kā piemēram profila ikona ir izmantotas šajā darbā. Tas ir pieejams <https://iconscout.com/>.

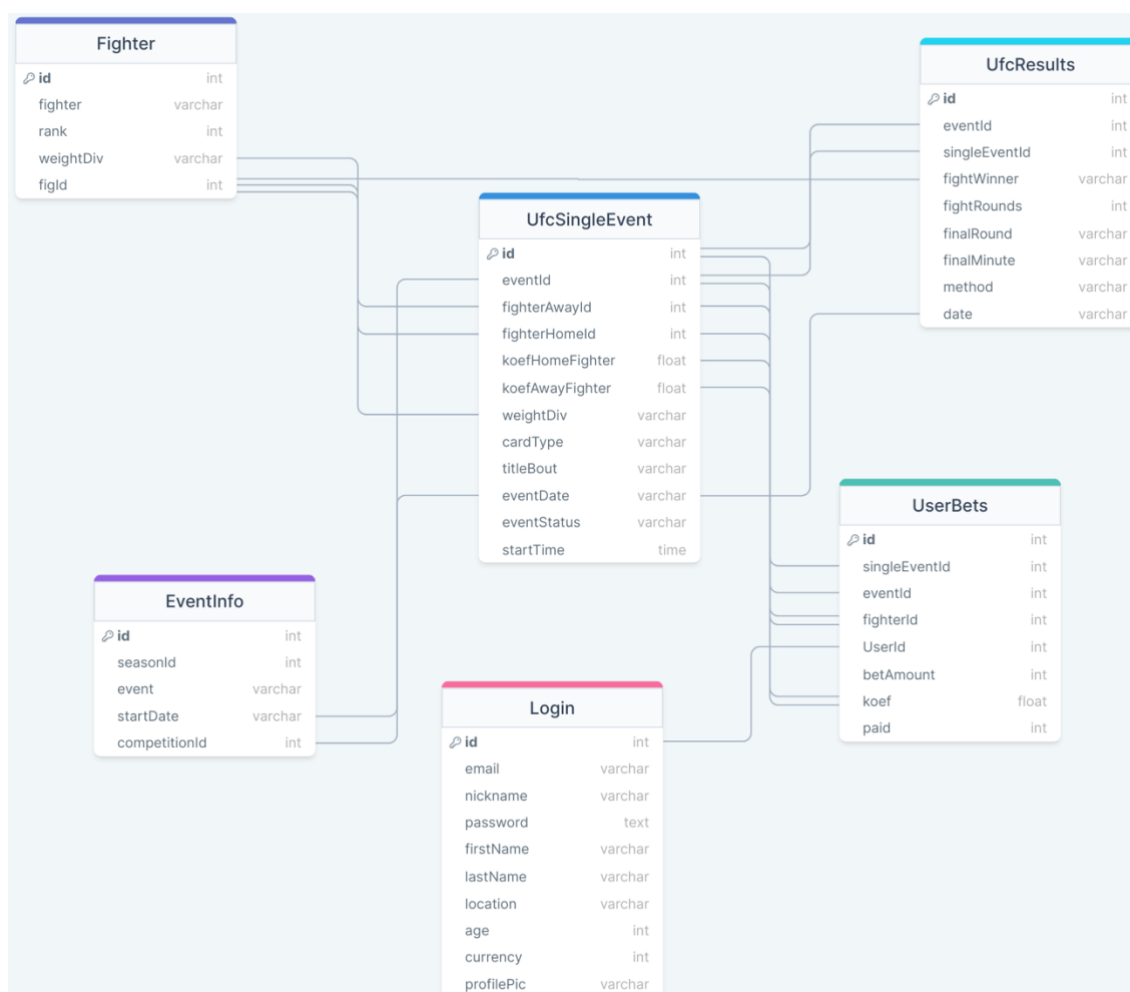
Sportradar – pasaulē vadošais sporta tehnoloģiju uzņēmums, kas atrodas sporta, mediju un derību krustpunktā. Šī kompānija nodrošina *API* par vairākiem sporta veidiem, piemēram, futbols, basketbols, bokss, krikets un daudz citu sporta veidu. No šiem sporta veidiem šī kompānija arī nodrošina *API* jauktās cīņas mākslas sportam. *API* ir lietojumprogrammas saskarne (angļu v. *Application programming interface*), kas tiek uzskatīts par programmatūras starpnieku. Tas ļauj divām lietojumprogrammām sazināties vienai ar otru. Tas ir pieejams veids, kā iegūt un koplietot datus organizācijās un starp tām. Tas ir pieejams <https://sportradar.com/>.

4. SISTĒMAS STRUKTŪRAS MODELIS

Šī tīmekļa vietne darbojas ar sešām datu tabulām:

- *Login* tabula - glabā informāciju par lietotāju.
- *EventInfo* tabula - glabā informāciju par cīņu pasākumiem.
- *UfcSingleEvent* tabula - glabā informāciju kopumu par visu pasākumu un cīkstoņiem, lai likmju lapa vietnē spētu strādāt.
- *Fighter* tabula - glabā informāciju par cīkstoņiem.
- *UfcResults* tabula - glabā informāciju par notikušām cīņām un to rezultātiem.
- *UserBets* tabula - glabā informāciju par lietotāja veiktajām likmēm.

Datu bāzu izkārtojumu un savienojanos var redzēt šajā tīmekļa vietnē <https://drawsql.app/teams/hrwells/diagrams/complete>. Šajā vietnē ir iespēja redzēt kā datu bāzes savstarpēji savienojas (skatīt 10. attēlu). Diagrammas veidošanas rīks pieejams: <https://drawsql.app/diagrams>.



10. attēls. Datu bāzu izkārtojums

Visas sešas datu tabulas glabā informāciju par cīņām un lietotājiem. Lielākā daļa tabulu savienojas ar tabulu *UfcSingleEvent*, kas glabā datus par katru cīņu no katra pasākuma (skatīt 3. tabulu).

3. tabula

Tabulas *UfcSingleEvent* datu struktūra

Nosaukums	Apraksts	Datu tips	Piezīmes
id	Viena pasākuma unikālais identifikators	Int(11)	Primārā atslēga AUTO_INCREMENT
eventId	Pasākuma identifikators	Int(11)	
fighterAwayId	Iebraucēja cīkstoņa identifikators	Int(11)	
fighterHomeId	Mājas cīkstoņa identifikators	Int(11)	
koefHomeFighter	Mājas cīkstoņa koeficients	float	
koefAwayFighter	Iebraucēja cīkstoņa koeficients	float	
weightDiv	Noteiktās cīņas svara kategorija	Varchar(50)	
cardType	Noteiktās cīņas konkursa tips (Galvenais, vidējais vai pēdējais)	Varchar(50)	
titleBout	Cīņas tips vai ir čempiona vai nav.	Varchar(50)	
eventDate	Pasākuma datums	Varchar(50)	
eventStatus	Pasākuma stāvoklis	Varchar(50)	
startTime	Pasākuma sākšanās laiks	time	

Tabula *Fighter* glabā datus par katru cīkstoni un tā vietu visā rangā viņa noteiktajā svara kategorijā. Svara kategorija tabulā *Fighter*, kas ir kolonna *weightDiv* padod datus tālāk uz tabulu *UfcSingleEvent* kolonnu *weightDiv*. No tabulas *Fighter* kolonnu *figId* padod tālāk uz 2 tabulām un to kolonnām, pirmā ir *UfcSingleEvent*, kur tās kolonnās *fighterAwayId* un *fighterHomeId* ievieto *figId* kolonnas datus, kurus izmanto, lai atšķirtu cīņas cīkstoņus. Otrā tabula kurā ievieto šos datus ir *UfcResults* un tās kolonnā *fightWinner* pēc kuras nosaka cīņas uzvarētāju (skatīt 4. tabulu).

4. tabula

Tabulas *Fighter* datu struktūra

Nosaukums	Apraksts	Datu tips	Piezīmes
id	Cīkstoņa unikālais identifikators	Int(11)	Primārā atslēga AUTO_INCREMENT
fighter	Cīkstoņa vārds, uzvārds	Varchar(100)	
rank	Cīkstoņa rangs	Int(11)	
weightDiv	Cīkstoņa svara kategorija	Varchar(100)	

figId	Cīkstoņa identifikators	Int(11)	
-------	----------------------------	---------	--

No tabulas *EventInfo* izvada 2 kolonnas *startDate* un *competitionId*, kuras ievieto tabulā *UfcSingleEvent*. Kolonnu *startDate* ievieto kolonnā *eventDate*, kas nosaka pasākuma sākuma datumu. Kolonnu *competitionId* ievieto kolonnā *eventId*, kas nodod noteikto identifikatoru cīņai, lai spētu sašķirot pasākumus pēc tā noteiktā identifikatora (skatīt 5. tabulu).

5. tabula

Tabulas *EventInfo* datu struktūra

Nosaukums	Apraksts	Datu tips	Piezīmes
id	Pasākuma unikālais identifikators	Int(11)	Primārā atslēga AUTO_INCREMENT
seasonId	Pasākuma sesijas identifikators	Int(11)	
event	Pasākuma vārds	Varchar(255)	
startDate	Pasākuma datums	Varchar(255)	
competitionId	Pasākuma konkursa identifikators	Int(11)	

No tabulas *Login* kolonnas *id* padod tālāk datus uz tabulu *UserBets* kolonnu *UserId*, kas izmanto šos datus, lai atšķirtu kurš lietotājs ir veicis likmi par noteikto cīkstoni izvēlētajā pasākuma un cīņā (skatīt 6. tabulu).

6. tabula

Tabulas *Login* datu struktūra

Nosaukums	Apraksts	Datu tips	Piezīmes
id	Lietotāja unikālais identifikators	Int(11)	Primārā atslēga AUTO_INCREMENT
email	Lietotāja epasts	Varchar(64)	
nickname	Lietotāja lietotājvārds	Varchar(30)	
password	Lietotāja parole	text	
firstName	Lietotāja vārds	Varchar(20)	
lastName	Lietotāja uzvārds	Varchar(30)	
location	Lietotāja atrašanās vieta	Varchar(30)	
age	Lietotāja vecums	Int(2)	Default(0)
currency	Lietotāja virtuālās naudas kopsumma	Int(11)	Default(1000)
profilePic	Lietotāja bilde	Varchar(255)	includes/profileImages/guest.png

UserBets tabula iegūst datus no tabulas *UfcSingleEvent*. Pirmā kolonna no tabulas *UfcSingleEvent* ir *id*, kuru ievieto tabulas *UserBets* kolonnā *singleEventId*, lai to izmantotu kā identifikatoru, kas atšķirtu noteiktā pasākuma cīņas. Kolonnā *eventId* ievieto *eventId* datus no *UfcSingleEvent* tabulas. *FighterId* kolonnā ievieto kolonnu *fighterAwayId* un *fighterHomeId* datus, kuri tiek izmantoti, lai noteiktu cīkstoņa unikālo identifikatoru.. Kolonnā *Koef* ievieto *koefAwayFighter* un *koefHomeFighter* kolonnu datus, kurus izmanto, lai noteiktu cīkstoņa koeficientu (skatīt 7. tabulu).

7. tabula

Tabulas *UserBets* datu struktūra

Nosaukums	Apraksts	Datu tips	Piezīmes
id	Likmes unikālais identifikators	Int(11)	Primārā atslēga AUTO_INCREMENT
singleEventId	Viena pasākuma identifikators	Int(11)	
eventId	Pasākuma identifikators	Int(11)	
fighterId	Cīkstoņa identifikators	Int(11)	
userId	Lietotāja identifikators	Int(11)	
betAmount	Vienas likmes noteiktais daudzums	Int(11)	
koef	Cīkstoņa koeficients	float	
paid	Pārbaudītājs, kas pārbauda vai lietotājam ir veikta izmaksa.	Int(1)	

UfcResults tiek padoti dati no 2 datu bāzes tabulām. Pirmā tabula no, kuras padod datus ir *Fighter* un tās kolonnas *figId* ievieto datus *fightWinner* kolonnā. Otrā tabula ir *UfcSingleEvent* no kuras tiek padotas 3 kolonnas ar datiem uz *UfcResults* kolonnām. Kolonna *eventId* padod datus uz kolonnu *eventId*. Nākamā kolonna ir *id*, kas padod datus tālāk uz *singleEventId*. Pēdējā kolonna, kas nodod datus uz *UfcResults* ir kolonna *eventDate*, kas ievieto datus kolonnā *date* (skatīt 8. tabulu).

8. tabula

Tabulas *UfcResults* datu struktūra

Nosaukums	Apraksts	Datu tips	Piezīmes
id	Pasākuma iznākuma unikālais identifikators	Int(11)	Primārā atslēga AUTO_INCREMENT

eventId	Pasākuma identifikators	Int(11)	
singleEventId	Viena pasākuma identifikators	Int(11)	
fightWinner	Cīņas uzvarētāja identifikators	Varchar(255)	
fightRounds	Cīņas kopējie raundi	Int(1)	
finalRound	Cīņas pēdējais raunds	Varchar(255)	
finalMinute	Cīņas pēdējās minūtes	Varchar(255)	
method	Cīņas uzvaras veids	Varchar(255)	
date	Cīņas datums	Varchar(255)	

Liela daļa no datiem tiek ievākti un ievietoti datu bāzē izmantojot API. Šie dati ir saistīti ar cīņu nobeiguma rezultātu, uzvarētāju un vispārēju informāciju par cīkstoni, cīņu un pasākumu.

Izveidotajā tīmekļa vietnē ir daudz datu savienojumi, kas savstarpēji savienojas un darbojas. Datu bāze *db.php* ir savienota kopā ar visiem pārējiem failiem, skatoties uz to, ka tas ir viens no svarīgākajiem savienojumiem datu struktūrā (skatīt 11. attēlu).

Pēc datu bāzes savienojuma sāk darboties *register.php*, kas atbild par reģistrāciju un pieslēgšanos kontam. Reģistrāciju un pieslēgšanos nodrošina *loginVerify.php* un *registration.php*, kas ievietot datus datu bāzē vai izvada kļūdu ziņojumus.

Kad lietotājs ir pieslēdzies sāk darboties *javascript* faili, kas ir *checkbox.js*, *slider.js* un *script.js*. Šie *javascript* faili tiek pievienoti *header.php* un *footer.php*, kas ir tīmekļa vietnes galvene un kājene.

Javascript fails *script.js* nodrošina katra ievadlauka darbību vietnē. Padarot to funkcionālāku un nepieprasot tīmekļa vietnes atjaunināšanu, lai ievietotu datus datu bāzē.

Pēc *javascript* failu nodrošināšanas un kājenes, galvenes pievienošanas tīmekļa vietnei, sāk darboties vietnes pēc lietotāja pieslēgšanās tīmekļa vietnei. *Javascript* palīdz darboties vairākām vietnēm ar datu padošanu vai arī to izmanto priekš priekšgala sistēmas uzlabošanas.

Lai tiktu noteikts lietotājs tiek izmantots fails *sessions.php*, kur glabā lietotāja sesiju funkcijas. Šis fails arī tiek izmantots, lai lietotājs varētu pārtraukt sesiju, nospiežot uz pogas iziet (angļu v. *EXIT*).

Pirmā vietne ir *index.php*, kurā lietotājs nonāk pieslēdzoties savam kontam. Sākulapa atrodas *index.php*, tāpēc lietotājs tur tiek pievienots pēc pieslēgšanās.

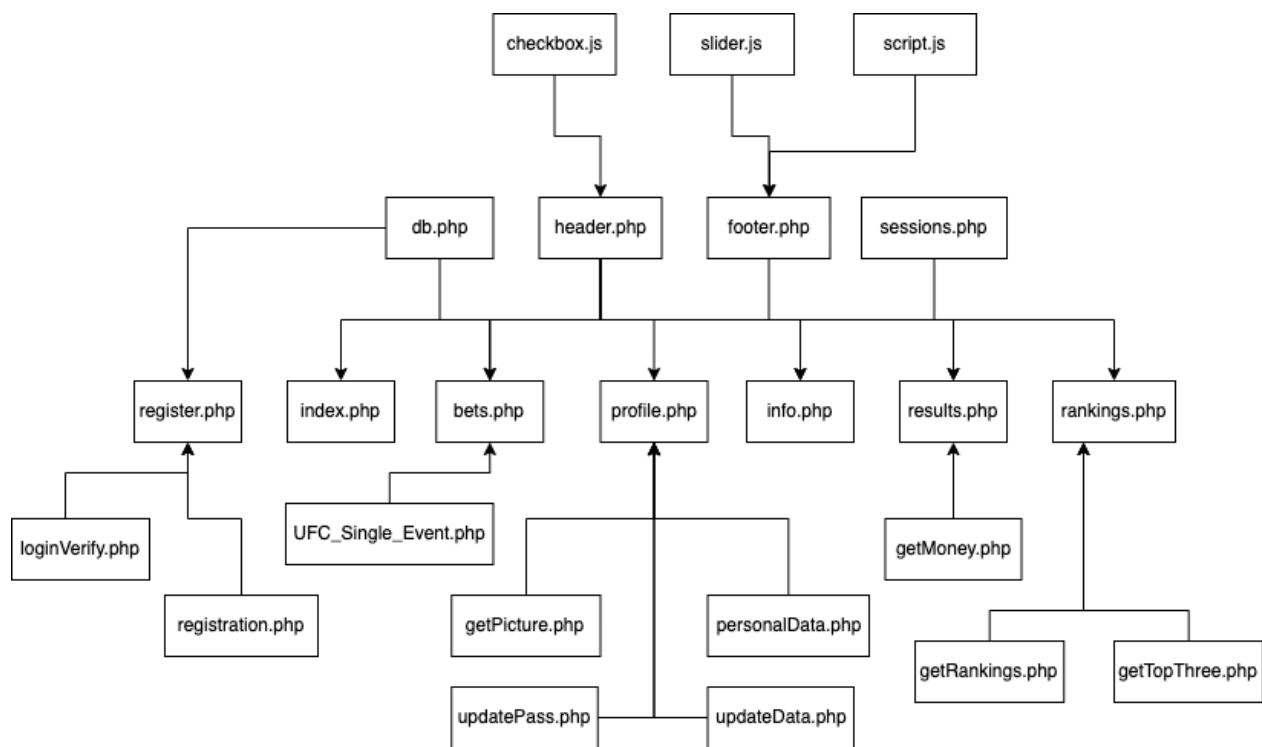
Otrā vietne ir *bets.php*, kur lietotājs var veikt likmes. Lai šīs likmes darbotos *bets.php* tiek savienots ar *UFC_Single_Event.php* kā arī šim failam darboties palīdz *checkbox.js*, kas nodrošina *javascript* datu nokļūšanu uz *UfcSingleEvent* failu.

Trešā vietne ir *profile.php* kurā lietotājs var savadīt informāciju vai atjaunināt to. Šis fails savienojas ar vairākiem failiem, kas palīdz funkcijām darboties šajā failā. Kad lietotājs pieslēdzas *profile.php* vietnei pirmā informācija ir ievadlauki, kas ir paredzēti priekš lietotāju datu savadīšanas, kurus nodrošina fails *personalData.php*. Ja lietotājs vēlas ievietot bildi tiek pievienots fails *getPicture.php*, kas ļauj bildi ievietot datu bāzē un pēc tam to attēlota lietotāja kontā. Šajā failā arī iekļauj *sessions.php*, lai varētu izvadīt lietotāja informāciju un saprast par, kuru lietotāju tiek ievadīta informācija. Ja lietotājs vēlas atjaunināt paroli tiek izmantots *updatePass.php*, kas atjaunina lietotāja informāciju datu bāzēs pēc paroles maiņas. Kā arī lietotājs var mainīt e-pasta adresi izmantojot *updateData.php*. Pēc e-pasta adreses atjaunināšanas lietotāja informācija datu bāzē tiek atjaunināta uz jauno e-pasta adresi.

Ceturtā vietne ir *info.php*, kur tiek attēlota informācija par vietni. Vietnē ir kastes ar informāciju, kurām tiek izmantots *slider.js*, lai bildes parādīto palēnināti, lēnām ieplūstot ekrānā.

Piektā vietne ir *results.php*, kur tiek izvadīti lietotāja likmju rezultāti par cīņu. Šis fails darbojas ar *getMoney.php* palīdzību, kas ievāc datus no datu bāzes par cīņu rezultātiem.

Sestā vietne ir *rankings.php*, kas ievieto visus lietotājus rangos un ievietot tos pakāpeniski to virtuālās summas daudzumam. Lai šis fails darbotos tiek piesaistīti 2 faili, kas ir *getRankings.php* un *getTopThree.php*. Lai secīgi novietotu pašus sekmīgākos lietotājus tiek izmantots *getTopThree.php*, kas iegūst 3 sekmīgākos lietotājus pēc virtuālās valūtas summas. Lai sekmētu pārējos 50 lietotājus, kas ir tikuši ranga sistēma tiek izmantots *getRankings.php*. Šis fails sekmē atlikušos 50 lietotājus ranga sistēmā pēc viņu virtuālās valodas summas. Diagramma veidošanas rīks pieejams: <https://app.diagrams.net/>.



11. attēls. Sistēmas komponentu modelis

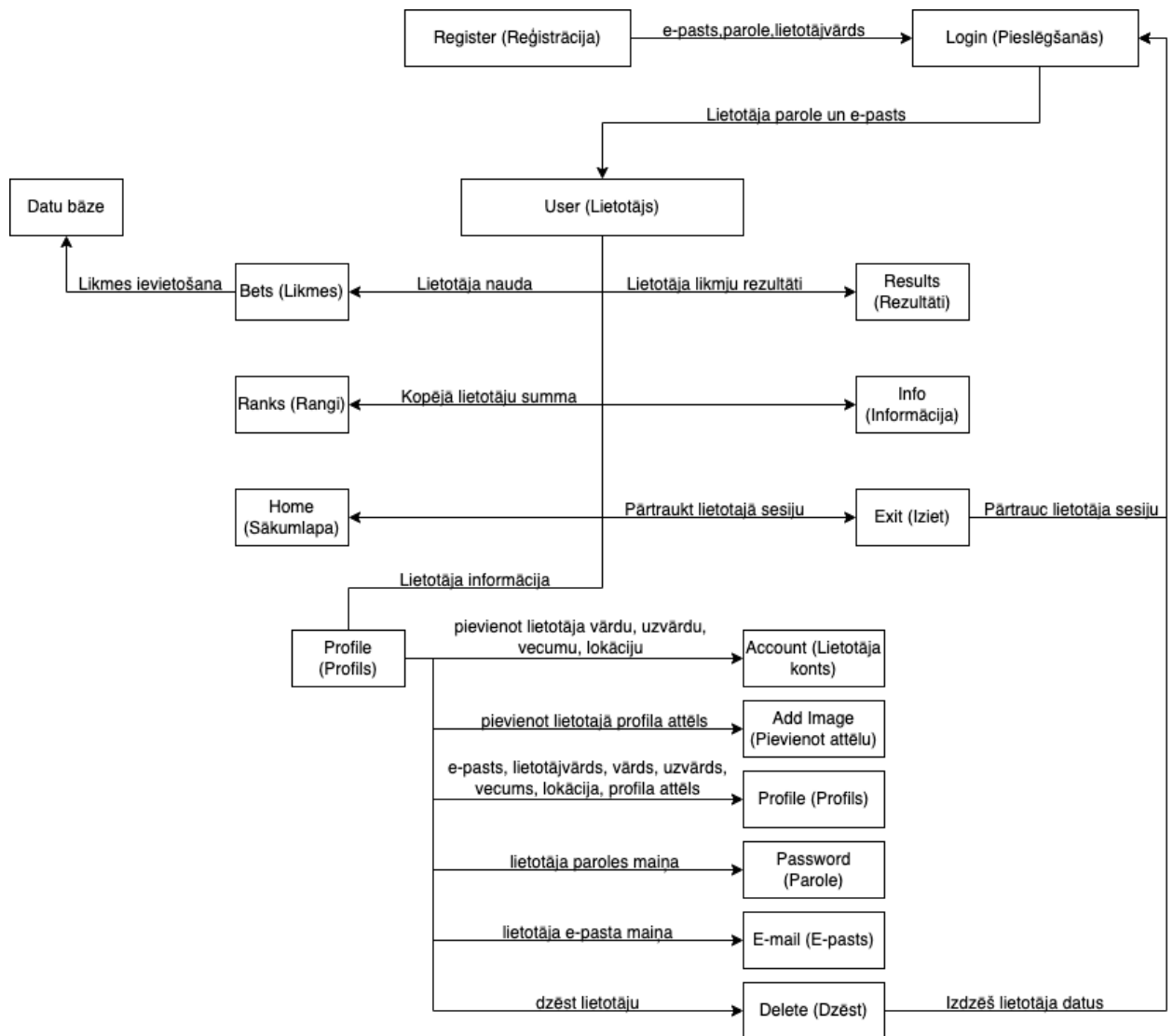
5. FUNKCIONĀLAIS UN DINAMISKAIS SISTĒMAS MODELIS

Datu plūsmas shēmā pirmā lietotāja kustība ir reģistrācija (angļu v. *Register*), kur ir jāievada nepieciešamie dati, lai lietotājs tiktu izveidots, kas ir - lietotājvārds, e-pasts, parole. Pēc reģistrācijas lietotājam ir pieejama opcija pieslēgšanās (angļu v. *Login*), kur ir nepieciešams ievadīt e-pastu un paroli, lai varētu pieslēgties vietnei.

Pēc pieslēgšanās lietotājam ir pieejamas 6 vietnes sekcijas. Sekcija Sākumlapa (angļu v. *Home*) lietotājam attēlo pirmo skatu, kuru lietotājs redz pieslēdzoties vietnei. Sekcija Likmes (angļu v. *Bets*) lietotājs var veikt likmes uz notiekošajām cīņām, kur pēc tām ievietotā likme tiek nogādāta datu bāzē, kura pēc cīņas beigām atgriež lietotājam naudu, ja cīņa ir bijusi veiksmīga. Sekcija Rangs (angļu v. *Ranks*) attēlo visu lietotāju kopējo summu un sarindo tos dilstošā secībā ar bagātāko lietotāju pirmajā vietā. Sekcija Rezultāti (angļu v. *Results*) attēlo lietotāja likmju rezultātus. Sekcija Informācija (angļu v. *Info*) attēlo skaidrojumu par programmas vispārējo funkcionalitāti un kādēļ tā tika radīta. Uzspiežot uz pogas Iziet (angļu v. *Exit*), lietotājam tiek pārtraukta sesija, un tas tiek nogādāt atpakaļ pie pieslēgšanās izvēlnes.

Sekcija Profils (angļu v. *Profile*) piedāvā izvēlni starp vairākām ievadīšanas opcijām. Izvēlnē Konts (angļu v. *Account*) lietotājam dod iespēju ievadīt informāciju par viņu pašu, piemēram, vārds, uzvārds, vecums un lokācija. Izvēlnē Pievienot bildi (angļu v. *Add Image*) ļauj lietotājam pievienot profila attēlu. Izvēlnē Profils (angļu v. *Profile*) attēlo lietotāja e-pastu, lietotājvārdu, vārdu, uzvārdu, vecumu, lokāciju un profila attēlu. Izvēlnē Parole (angļu v. *Password*) ļauj lietotājam mainīt esošo paroli pret citu. Izvēlnē E-pasts (angļu v. *E-mail*) ļauj lietotājam mainīt esošo e-pastu pret citu. Izvēlnē Dzēst (angļu v. *Delete*) piedāvā lietotājam opciju dzēst lietotāja datus un atgriež to pie pieslēgšanās izvēlnes (Tutorialspoint, 2023).

Attēla tiek attēlota informācija par lietotāja datu plūsmu kustoties pāri tīmekļa vietnei. Katra kustība pārnes lietotāja datus, kas ievieto vai izvada tos. (skatīt 12. attēlu).



12. attēls. Lietotāja datu plūsma

6. DATU STRUKTŪRAS APRAKSTS

Es izmantoju vesela skaitļa (angļu v. *integer*) tipa mainīgo sazarojuma nosacījumam (angļu v. *if statement*). Tas nosaka lietotāja vecumu un, ja ievadītais ir zemāks par 18, izvada paziņojumu. Vesels skaitlis programmēšanā ir datu tips, kuru izmanto, lai attēlotu reālus skaitļus, kuriem nav daļskaitļu vērtības. Veselu skaitļa tipus izmanto vairākos gadījumos, piemēram, kāda cikla veidošanā ar specifisku nosacījumu izmantojot veselu skaitli vai arī mainīga un datu masīva definēšanā (Rouse, 2016).

Citas pārbaudes, piemēram, *if* nosacījuma pārbaude priekš ievadlaukiem, iekļauj simbolu virknes mainīgo (angļu v. *string*). Tas pārbauda, vai ievadlauki ir tukši, ja ir, tad izvada attiecīgo ziņojumu. *If* nosacījums ir programmēšanas nosacījuma apgalvojums, kas, ja ir patiess, izpilda kāda veida funkciju vai parāda informāciju (Computer Hope, 2019).

Priekš lietotāja vārda, uzvārda un lietotājvārda izvades un ievades es izmantoju simbolu virknes mainīgo (angļu. *string*). Simbolu virknes mainīgos izmanto, lai glabātu tekstu un rakstzīmes. Šie mainīgie veido rakstzīmju virknes, kad tās izsauc ar iekļauto vērtību (Busbee un Braunschweig, 2018).

Pie lietotāja konta iestatījumiem, tiek piedāvāta valsts izvēle, kur tiek izmantots datu masīvs (angļu v. *array*). Šajā datu masīvā glabā valstu nosaukumus. Masīvs ir vienu datu tipu kopa, kas glabājas masīva atmiņā. Tas atvieglo vairāku datu tipu izsaukšanu un aprēķināšanu, vienkārši pievienojot nobīdi bāzes vērtībai (Geeksforgeeks, 2023).

Priekš ranga izvadišanas augstākajām trīs vietām izmantoju *foreach* ciklu, kur izmanto vesela skaitļa tipa mainīgos. *Foreach* ir cikls, kas darbojas gan uz masīviem, gan objektiem. Tas dublē tā dotos datus, kamēr tiek padoti dati uz to. Pārsvārā šo ciklu izmanto, lai ietaupītu atmiņu un padarītu efektīvāku un optimālāku kodu un pašu programmu (Vats, 2020).

Priekš katra cīkstoņa ir noteikts koeficients, šie koeficienti izmanto *float* datu tipu, lai glabātu informāciju. *Float* apzīmē reālus skaitļus tāpat kā *integer*. Tiem atšķirīgs ir tas, ka *float* var glabāt datus ar komatu, tas nozīmē, ka *float* var glabāt decimāldaļu skaitļus (Computer Science Wiki, 2019).

Lai atkārtoti veidotos likmju sadaļā kastes ar cīkstoņu informāciju un likmju opcijām, tiek izmantots *while* cikls, lai turpinātu saukt kastes, kamēr ir dati, no kuriem tās veidot. *While* cikls turpina izpildīt kodu, kamēr noteiktais nosacījums ir patiess. Šī cikla vērtība tiek pārbaudīta katru reizi, kad tā tiek izsaukta, ja pat tā tiktu mainīta cikla vidū, tā neapstāsies, kamēr nebeigsies datu, ko izvadīt. Ja cikla vērtība jau no paša sākuma ir nepatiesa, tad šis cikls neizpildīsies pat 1 reizi (Php, 2023).

Kad lietotājs ir pieslēdzies vietnei, tiek izmantota funkcija *session_start()*, lai uzsāktu lietotāja sesiju tīmekļa vietnē. Sesiju uzsākšana ir veids, kā īslaicīgi uzglabāt datus un padarīt tos pieejamus. Sesija ļauj tīmekļa vietnes serverim iegūt informāciju par lietotāju un viņa darbību. Kad sesiju uzsāk tas izveido konkrētu identifikatoru sesijai. Vietni aizverot, tiek pārtraukta sesija (Simplilearn, 2023).

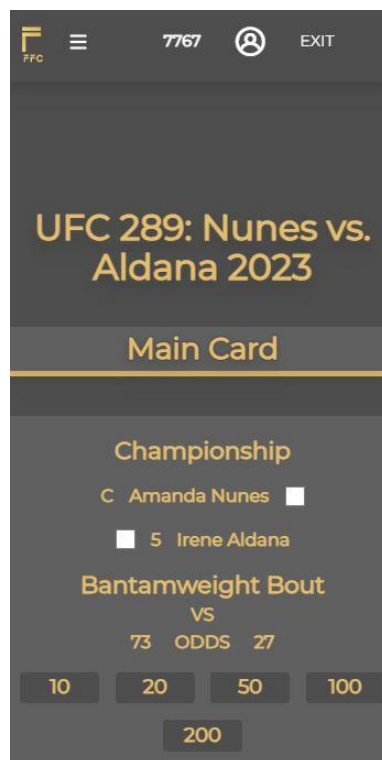
7. REAGĒJOŠA TĪMEKĻA VIETNE

Reagējoša tīmekļa vietne ir svarīga, lai tā varētu tikt izmantota uz visām ierīcēm. Katra vietne tiek apstrādāta attiecīgi ekrānu izmēriem. Skatoties uz to, ka lietotāji izmanto ne tikai datoru vai mobilās ierīces, tā tiek izstrādāta visām ierīcēm, lai dizains un lapas novietojums būtu ērts izmantošanai un katra attēlota informācija būtu viegli saskatāma un salasāma. Galvenais mērķis reagējošas vietnes izstrādē bija tāds, lai lietotājs varētu izmantot to jebkurā vietā, jebkurā laikā.

7.1. MAZIE EKRĀNI

Lai šī tīmekļa vietne būtu izmantojama jebkur un jebkad, tā tika izstrādāta uz visiem ekrāniem. Specifiski tā tika izstrādāta skatoties uz to, ka liela daļa lietotāju izmanto mobilo tālruni ikdienā un ar to ir vieglāka pieejamība tīmekļa vietnei.

Primārā sekcija par, kuru tik piedomāts bija paša likmju vietne, lai lietotājs spētu veikt likmes tur ātri, viegli un varētu saprast izvadīto informāciju (skatīt 13. attēlu).



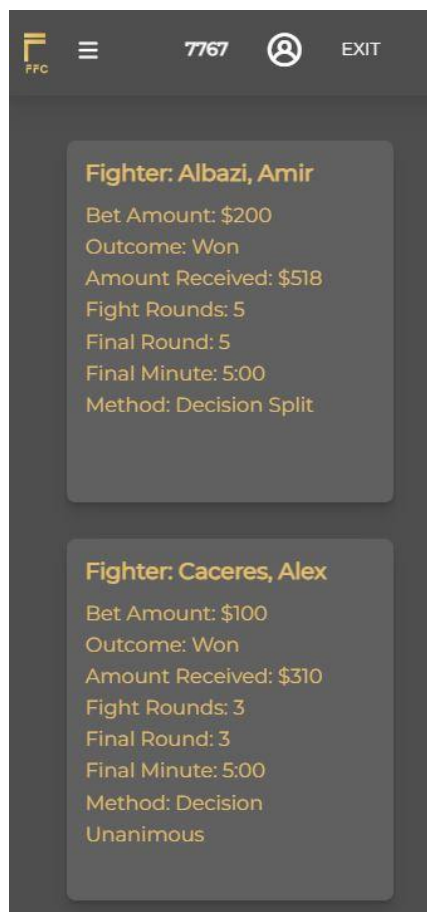
13. attēls. Mazo ekrānu likmju sadaļa

Kad lietotājs vēlas veikt konta konfigurāciju, ir jādodas uz konta iestatījumu sadaļa tīmekļa vietnē. Šī sadaļa arī tika piemērota maziem ekrāniem (skatīt 14. attēlu).

The screenshot shows a mobile application interface for account settings. The header is dark with the PFC logo, a menu icon, the number 7767, a user icon, and an EXIT button. A left sidebar menu contains options: Menu, Account, Add Image, Profile, Password, and E-mail. The main area is titled 'Account Settings' and contains input fields for First Name (Ralfs), Last name (Dambitis), Location (Latvia), and Age (20). A Done button is at the bottom.

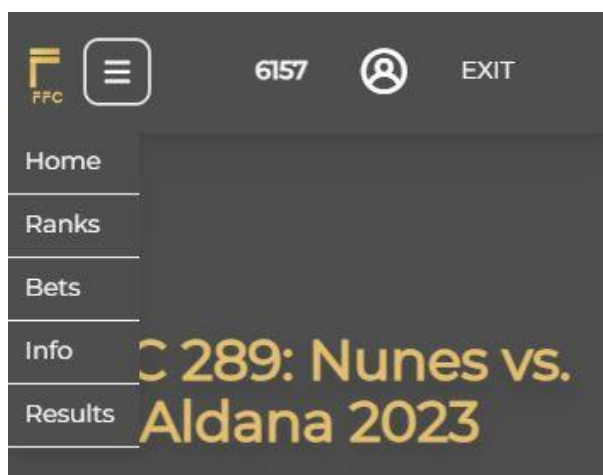
14. attēls. Mazo ekrānu konta sadaļa

Lai lietotājs varētu redzēt likmju rezultātus viņam jādodas uz rezultātu (angļu v. *Results*) vietni. Šī vietne tika piemērota maziem ekrāniem attiecīgi ar to, lai lietotājs spētu saredzēt cīņas statistiku, iegūto naudu un cīņas rezultātu (skatīt 15. attēlu).



15. attēls. Mazo ekrānu rezultātu sadaļa

Navigācijas josla uz maziem ekrāniem pārveidojas par hamburgera izskata navigāciju. Tā pārtop par pogu, uz kuras uzspiežot atveras navigācijas josla ar visām hipersaitēm novietotām vertikāli (skatīt 16. attēlu).

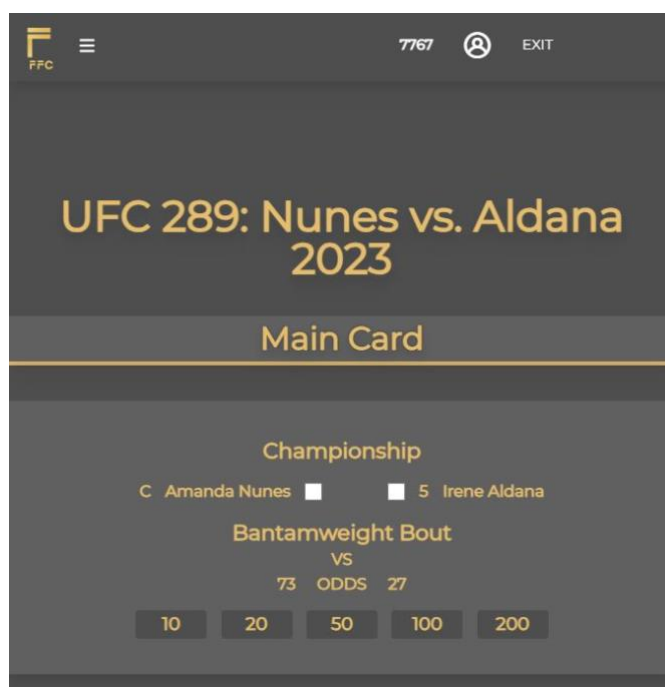


16. attēls. Mazo ekrānu rezultātu sadaļa

7.2. VIDĒJIE EKRĀNI

Nākamie reaģējošie ekrāni ir vidējie ekrāni, tostarp ir planšetdatori, portatīvie datori un lielāku izmēru televizori. Pie šiem ekrāniem nebija grūti strādāt skatoties, ka tie ir mazliet mazāki par lielajiem, bet jeb kurā gadījumā uzlabojumi tika veikti. Katra no šīm vietnēm tika pielāgotas ekrāna izmēram.

Pirmā sadaļa, kas tika uzlabota ir likmju sadaļa. Skatoties uz to, ka izmērā tā neatspoguļoja pārāk daudz no lielajiem ekrāniem, vienīgie uzlabojumi, kuri tika veikti bija attēlu noņemšana un burtu samazināšana (skatīt 17. attēlu).



17. attēls. Vidējo ekrānu likmju sadaļa

Kad lietotājs vēlas veikt konta konfigurāciju, ir jādodas uz konta iestatījumu sadaļa tīmekļa vietnē. Šī sadaļa arī tika piemērota vidējiem ekrāniem. Viena no izmaiņām, kas tika veikta šajā sadaļā bija ievadlauku novietojums. Kā arī burtu izmēri tika mainīti (skatīt 18. attēlu).

FFC

7767

EXIT

Menu

Account

Add Image

Profile

Password

E-mail

Account Settings

First Name

Ralfs

Last name

Dambitis

Location

Latvia

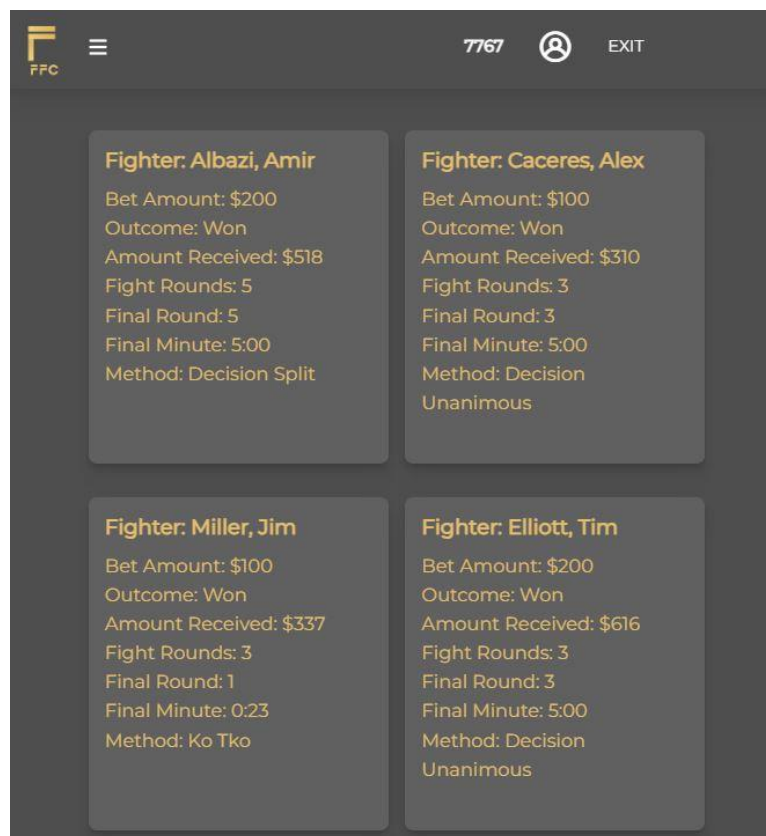
Age

20

Done

18. attēls. Vidējo ekrānu konta sadaļa

Lai lietotājs varētu redzēt likmju rezultātus viņam jādodas uz rezultātu (angļu v. *Results*) vietni. Šī vietne tika piemērota vidējam ekrānam attiecīgi ar to, lai lietotājs spētu saredzēt cīņas statistiku, iegūto naudu un cīņas rezultātu. Protams, uz vidējiem ekrāniem var saredzēt vairāk nekā mazajiem tādēļ tiek attēlotas vairāk kastītes (skatīt 19. attēlu).

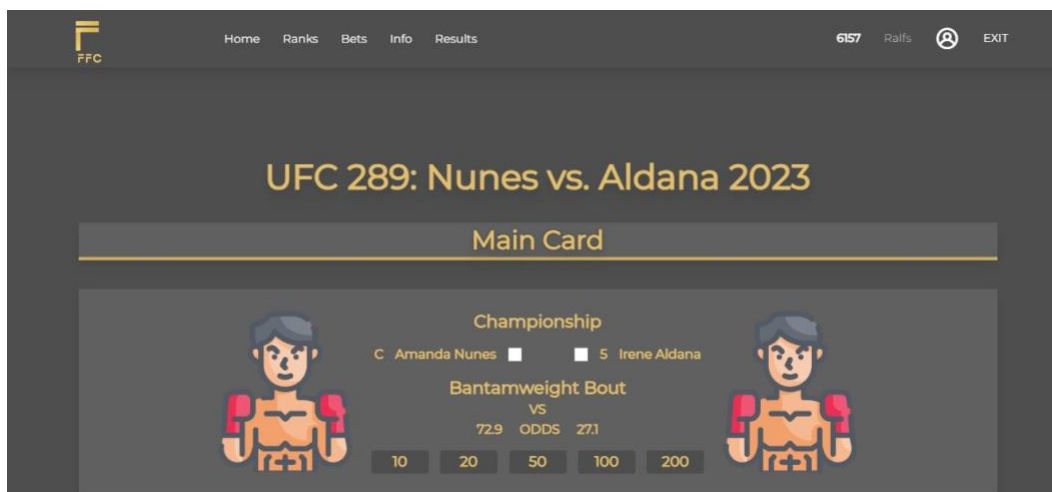


19. attēls. Vidējo ekrānu rezultātu sadaļa

7.3. LIELIE EKRĀNI

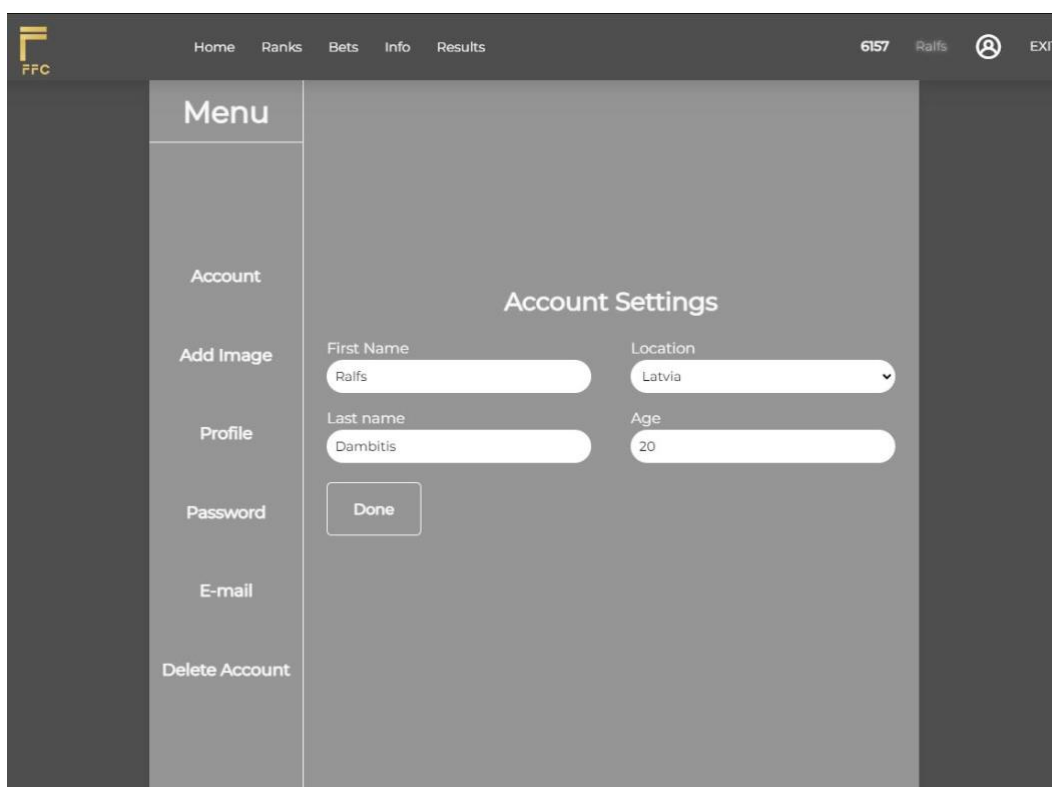
Nākamie reaģējošie ekrāni ir lielle ekrāni. Lielos ekrānus uzskata par televizoriem un monitoriem. Pie šiem ekrānu izmēriem bija vienkārši strādāt, skatoties uz to, ka tīmekļa vietni veidoju uz tiem. Vienīgā problēma ar lieliem ekrāniem ir izstrādāt pietiekami lielus burtus nesalaužot to novietojumu. Katrām no minētajām vietnēm ir piemērots attiecīgais izmērs.

Pirmā sadaļa, kas tika uzlabota ir likmju sadaļa. Lielie ekrāni ir vienkāršākā sadaļa, kuru padarīt reaģējošu un tādēļ arī uz lielajiem ekrāniem tiek pievienotas bildes, kas padara šos laukumus pievilcīgākus. Kā arī var redzēt to, ka navigācijas josla ir atgriezusies lapas augšā un ir novietota horizontāli (skatīt 20. attēlu).





20. attēls. Lielo ekrānu likmju sadaļa

Kad lietotājs vēlas veikt konta konfigurāciju, ir jādodas uz konta iestatījumu sadaļa tīmekļa vietnē. Šī sadaļa arī tika piemērota lielajiem ekrāniem. Ievad lauki tagad ir novietoti citā secībā, horizontāli. Kā arī vispārējie kastes lauki ir samazināti un tiem sānos ir iedotas apmales (skatīt 21. attēlu).



21. attēls. Lielo ekrānu konta sadaļa

Lai lietotājs varētu redzēt likmju rezultātus viņam jādodas uz rezultātu (angļu v. *Results*) vietni. Šī vietne tika piemērota lielajiem ekrāniem. Skatoties uz to, ka ekrāns ir stipri lielāks izvada vairāk kastes ārā uz ekrāna ar rezultātu informāciju. Kā arī dati ir vieglāk pārredzami (skatīt 22. attēlu).

	Home Ranks Bets Info Results	7767	Ralfs  EXIT
Fighter: Albazi, Amir Bet Amount: \$200 Outcome: Won Amount Received: \$518 Fight Rounds: 5 Final Round: 5 Final Minute: 5:00 Method: Decision Split	Fighter: Caceres, Alex Bet Amount: \$100 Outcome: Won Amount Received: \$310 Fight Rounds: 3 Final Round: 3 Final Minute: 5:00 Method: Decision Unanimous	Fighter: Miller, Jim Bet Amount: \$100 Outcome: Won Amount Received: \$337 Fight Rounds: 3 Final Round: 1 Final Minute: 0:23 Method: Ko Tko	Fighter: Elliott, Tim Bet Amount: \$200 Outcome: Won Amount Received: \$616 Fight Rounds: 3 Final Round: 3 Final Minute: 5:00 Method: Decision Unanimous

22. attēls. Lielo ekrānu rezultātu sadaļa

8. TESTĒŠANA

Testēšana ir process, kurā tiek novērtēts un pārbaudīts, vai testējamais programmatūras produkts vai lietojumprogramma pilda iecerētās funkcijas un darbības. Testēšana ietver kļūdu novēršanu, izstrādes optimizāciju, veiktspējas uzlabošanu un programmas noslodzes atvieglošanu. Testēšana ir svarīgs process, lai saglabātu un uzlabotu produkta kvalitāti. Galvenais mērķis testēšanā ir veikt validāciju un verifikāciju. Šo procesu rezultātā var secināt programmas kļūmes, lai varētu veikt labojumus un novērst defektus (IBM, 2023). Saite uz *Github* kodu <https://github.com/HighRollerLV/FantasyFightingChampionship.git>.

9. tabula

Testu veikšana un to rezultātu tabula

Nr .p. k.	Testējamā funkcija	Scenārijs	Iecerētais rezultāts	Iegūtais rezultāts
1.	Lietotāja reģistrācija.	Lietotājs korekti un precīzi aizpilda reģistrācijas ievadlaukus.	Zem reģistrācijas formas izvada paziņojumu <i>Dati ir pievienoti veiksmīgi!</i> (angļu v. <i>The data has been added successfully!</i>) .	Zem reģistrācijas formas izvada paziņojumu <i>Dati ir pievienoti veiksmīgi!</i> (angļu v. <i>The data has been added successfully!</i>) (skatīt 47. attēlu).
2.	Lietotāja reģistrācija ar nepareizu e-pastu.	Lietotājs neprecīzi ieraksta e-pastu reģistrācijas e-pasta ievadlaukā.	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Lūdzu ievadiet pareizi e-pasta adresi</i> (angļu v. <i>Please enter a valid email address</i>).	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Lūdzu ievadiet pareizi e-pasta adresi</i> (angļu v. <i>Please enter a valid email address</i>) (skatīt 25. attēlu).
3.	Lietotāja reģistrācija ar tukšiem ievadlaukiem.	Lietotājs neaizpilda nevienu ievadlauku reģistrācijas sekcijā.	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Aizpildiet visus ievadlaukus! Lūdzu mēģiniet vēlreiz!</i> (angļu v. <i>All fields are required! Please try again!</i>).	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Aizpildiet visus ievadlaukus! Lūdzu mēģiniet vēlreiz!</i> (angļu v. <i>All fields are required! Please try again!</i>) (skatīt 23. attēlu).
4.	Lietotāja reģistrācija ar vienu tukšu ievadlauku.	Lietotājs neaizpilda vienu ievadlauku reģistrācijas sekcijā.	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Aizpildiet visus ievadlaukus! Lūdzu mēģiniet vēlreiz!</i> (angļu v. <i>All fields are required! Please try again!</i>).	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Aizpildiet visus ievadlaukus! Lūdzu mēģiniet vēlreiz!</i> (angļu v. <i>All fields are required! Please try again!</i>) (skatīt 26. attēlu).

5.	Lietotāja reģistrācija ar nedrošu paroli.	Lietotājs mēģina reģistrēties ar nedrošu paroli.	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Parolei jāsaturs vismaz 8 rakstzīmes, tai skaitā vismaz viens cipars un viens lielais burts!</i> (angļu v. <i>Password must contain at least 8 characters, including at least one number and one capital letter!</i>).	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Parolei jāsaturs vismaz 8 rakstzīmes, tai skaitā vismaz viens cipars un viens lielais burts!</i> (angļu v. <i>Password must contain at least 8 characters, including at least one number and one capital letter!</i>). (skatīt 24. attēlu).
6.	Lietotāja reģistrācija ar atšķirīgām parolēm.	Lietotāja paroles nesakrīt paroles un atkārtotās paroles ievadlaukos.	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Paroles nesakrīt! Lūdzu mēģiniet vēlreiz!</i> (angļu v. <i>Passwords do not match! Please try again!</i>).	Zem reģistrācijas formas izvada kļūdas ziņojumu <i>Paroles nesakrīt! Lūdzu mēģiniet vēlreiz!</i> (angļu v. <i>Passwords do not match! Please try again!</i>)(skatīt 27. attēlu).
7.	Lietotāja pieslēgšanās .	Lietotājs korekti un precīzi aizpilda pieslēgšanās ievadlaukus.	Pēc pieslēgšanās lietotāju pārvieto uz sāukmlapu, kur labajā sānā lejā izlec paziņojums <i>Tu esi veiksmīgi pieslēdzies!</i> (angļu v. <i>You have successfully logged in!</i>).	Pēc pieslēgšanās lietotāju pārvieto uz sāukmlapu, kur labajā sānā lejā izlec paziņojums <i>Tu esi veiksmīgi pieslēdzies!</i> (angļu v. <i>You have successfully logged in!</i>) (skatīt 49. attēlu, 50. attēlu).
8.	Lietotāja pieslēgšanās ar tukšiem ievadlaukiem.	Lietotājs mēģina pieslēgties ar tukšiem ievadlaukiem.	Zem pieslēgšanās formas izvada kļūdas paziņojumu <i>Parole vai E-pasts ir nepareizs, lūdzu mēģināt vēlreiz!</i> (angļu v. <i>Password or Email is incorrect, please try again!</i>).	Zem pieslēgšanās formas izvada kļūdas paziņojumu <i>Parole vai E-pasts ir nepareizs, lūdzu mēģināt vēlreiz!</i> (angļu v. <i>Password or Email is incorrect, please try again!</i>) (skatīt 28. attēlu).
9.	Pieslēgšanās ar neeksistējošu lietotāju.	Lietotājs mēģina pieslēgties vietnei aizpildot ievadlaukus ar neeksistējoša lietotāja datiem.	Zem pieslēgšanās formas izvada kļūdas ziņojumu <i>Lietotājs neeksistē</i> (angļu v. <i>User does not exist</i>).	Zem pieslēgšanās formas izvada kļūdas ziņojumu <i>Lietotājs neeksistē</i> (angļu v. <i>User does not exist</i>) (skatīt 29. attēlu).
10.	Lietotāja pieslēgšanās ar nepareiziem ievadlaukiem.	Lietotājs mēģina pieslēgties ar nepareiziem ievadlaukiem.	Zem pieslēgšanās formas izvada kļūdas paziņojumu <i>Parole vai E-pasts ir nepareizs, lūdzu mēģināt vēlreiz!</i> (angļu v. <i>Password or Email is incorrect, please try again!</i>).	Zem pieslēgšanās formas izvada kļūdas paziņojumu <i>Parole vai E-pasts ir nepareizs, lūdzu mēģināt vēlreiz!</i> (angļu v. <i>Password or Email is incorrect, please try again!</i>) (skatīt 28. attēlu).

11.	Lietotāja cīkstoņa atzīmēšana.	Lietotājs izvēlas cīkstoni starp diviem izvēles rūtiņu laukiem.	Izvēles rūtiņa atzīmējas, kad lietotājs izvēlas sevis izvēlēto cīkstoni.	Izvēles rūtiņa atzīmējas, kad lietotājs izvēlas sevis izvēlēto cīkstoni (skatīt 31. attēlu).
12.	Lietotājs veic likmi.	Lietotājs izvēlas cīkstoni un summu, kādu tiks uz cīkstoņa.	Paziņojums vietnes labā sāna augšdaļā parādās. Paziņojot, ka lietotājs ir veicis likmi veiksmīgi (angļu v. <i>Success! You have successfully placed a bet!</i>). Kā arī pārējās likmju opcijas tiek aizvērtas un ikona sarkana uzlec virsū bloķējot tās.	Paziņojums vietnes labā sāna augšdaļā parādās. Paziņojot, ka lietotājs ir veicis likmi veiksmīgi (angļu v. <i>Success! You have successfully placed a bet!</i>). Kā arī pārējās likmju opcijas tiek aizvērtas un ikona sarkana uzlec virsū bloķējot tās (skatīt 64. attēlu, 65. attēlu).
13.	Lietotājs veic likmi neatzīmējot cīkstoni.	Lietotājs neizvēloties cīkstoni mēģina veikt kādu no likmes opcijām.	Paziņojums vietnes labā sāna augšdaļā parādās. Paziņojot, ka lietotājam ir jāizvēlas cīkstonis, lai veiktu veiksmīgu likmi (angļu v. <i>Attention! Please choose a fighter.</i>).	Paziņojums vietnes labā sāna augšdaļā parādās. Paziņojot, ka lietotājam ir jāizvēlas cīkstonis, lai veiktu veiksmīgu likmi (angļu v. <i>Attention! Please choose a fighter.</i>) (skatīt 31. attēlu).
14.	Lietotājs veic likmi bez virtuālās naudas.	Lietotājs bez virtuālās naudas mēģina veikt likmi par cīņu	Paziņojums vietnes labā sāna augšdaļā parādās. Ziņojot lietotājam, ka tam nav pietiekami līdzekļu, lai veiktu veiksmīgu likmi <i>Kļūda! Jums nav pietiekami daudz līdzekļu, lai veiktu likmi!</i> (angļu v. <i>Error! You do not have enough funds to place a bet!</i>).	Paziņojums vietnes labā sāna augšdaļā parādās. Ziņojot lietotājam, ka tam nav pietiekami līdzekļu, lai veiktu veiksmīgu likmi <i>Kļūda! Jums nav pietiekami daudz līdzekļu, lai veiktu likmi!</i> (angļu v. <i>Error! You do not have enough funds to place a bet!</i>) (skatīt 32. attēlu).
15.	Lietotājs pievieno datus profila konta sadaļā.	Lietotājs aizpilda visus ievadlaukus precīzi un akurāti pie profila konta sadaļas.	Paziņojums zem pogas <i>Pabeigt</i> (angļu v. <i>Done</i>) parādās ar ziņu <i>Dati ir pievienoti veiksmīgi!</i> (angļu v. <i>Data has been added successfully!</i>).	Paziņojums zem pogas <i>Pabeigt</i> (angļu v. <i>Done</i>) parādās ar ziņu <i>Dati ir pievienoti veiksmīgi!</i> (angļu v. <i>Data has been added successfully!</i>) (skatīt 54. attēlu).
16.	Tukši ievadlauki profila konta sadaļā.	Lietotājs aizmirst aizpildīt vienu vai vairākus ievadlaukus.	Paziņojums zem pogas <i>Pabeigt</i> (angļu v. <i>Done</i>) parādās ar ziņu <i>Visi laukumi ir jāaizpilda!</i> (angļu v. <i>All fields must be filled!</i>).	Paziņojums zem pogas <i>Pabeigt</i> (angļu v. <i>Done</i>) parādās ar ziņu <i>Visi laukumi ir jāaizpilda!</i> (angļu v. <i>All fields must be filled!</i>) (skatīt 34. attēlu).

17.	Ievadlaukā ievadīts nepareizs vecums profila konta sadaļā.	Lietotājs ievadlaukā “Vecums” (angļu v. “Age”) ievada vecumu zemāku par 18.	Paziņojums zem pogas <i>Pabeigt</i> (angļu v. <i>Done</i>) parādās ar ziņu <i>Tev ir jābūt 18 gadus vecam vai vecākam!</i> (angļu v. <i>You have to be 18 years old or older!</i>).	Paziņojums zem pogas <i>Pabeigt</i> (angļu v. <i>Done</i>) parādās ar ziņu <i>Tev ir jābūt 18 gadus vecam vai vecākam!</i> (angļu v. <i>You have to be 18 years old or older!</i>) (skatīt 33. attēlu).
18.	Pievieno attēlu profila sadaļā pievienot attēlu.	Lietotājs faila ievadlaukā izvēlas attēlu un to pievieno kontam savam.	Pēc pogas <i>Pievienot</i> (angļu v. <i>Add</i>) nospiešanas bildi pievieno lietotāja kontam un attēlo pie sadaļas <i>Profils</i> (angļu v. <i>Profile</i>).	Pēc pogas <i>Pievienot</i> (angļu v. <i>Add</i>) nospiešanas bildi pievieno lietotāja kontam un attēlo pie sadaļas <i>Profils</i> (angļu v. <i>Profile</i>) (skatīt 35. attēlu, 56. attēlu).
19.	Profila paroles sadaļā atjaunina paroli.	Lietotājs akurāti un precīzi aizpilda ievadlaukus.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Jūsu parole ir veiksmīgi atjaunināta</i> (angļu v. <i>Your password has been updated successfully</i>).	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Jūsu parole ir veiksmīgi atjaunināta</i> (angļu v. <i>Your password has been updated successfully</i>) (skatīt 57. attēlu).
20.	Profila paroles sadaļā ievada nepareizu paroli.	Lietotājs savu neakurāti ieraksta savu pašreizējo paroli.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Jūsu pašreizējā parole ir nepareizi ievadīti. Mēģini vēlreiz.</i> (angļu v. <i>Your current password is incorrect. Try again</i>).	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Jūsu pašreizējā parole ir nepareizi ievadīti. Mēģini vēlreiz.</i> (angļu v. <i>Your current password is incorrect. Try again</i>) (skatīt 39. attēlu).
21.	Profila paroles sadaļā atstāj tukšu ievadlauku.	Lietotājs mainot paroli atstāj kādu no trīs ievadlaukiem tukšu.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Lūdzu ievadiet savu jauno paroli</i> (angļu v. <i>Please enter your new password</i>).	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Lūdzu ievadiet savu jauno paroli</i> (angļu v. <i>Please enter your new password</i> .) (skatīt 38. attēlu).
22.	Profila parole sadaļā jaunā parole sakrīt ar veco.	Lietotājs mainot savu paroli ievada jauno paroli vienādu ar veco paroli.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Jūsu jaunā parole sakrīt ar veco paroli. Lūdzu</i>	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Jūsu jaunā parole sakrīt ar veco paroli. Lūdzu</i>

			<i>izvēlieties citādāku paroli. (angļu v. Your new password is the same as your old password. Please choose a different password).</i>	<i>izvēlieties citādāku paroli. (angļu v. Your new password is the same as your old password. Please choose a different password) (skatīt 37. attēlu).</i>
23.	Profila paroles sadaļā paroles nesakrīt.	Lietotājs ievadot ievadlaukā jauno paroli, ievada citādāku paroli atkārtotās paroles ievadlaukā.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādas paziņojums <i>Jūsu paroles nesakrīt. Mēģiniet vēlreiz.</i> (angļu v. <i>Passwords do not match. Try again</i>).	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādas paziņojums <i>Jūsu paroles nesakrīt. Mēģiniet vēlreiz.</i> (angļu v. <i>Passwords do not match. Try again</i>) (skatīt 40. attēlu).
24.	Profila paroles sadaļā nedroša parole.	Lietotājs ievadot paroles, ievada nedrošu jauno paroli.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādas paziņojums <i>Parolei jāsaturs vismaz 8 rakstzīmes, tai skaitā vismaz viens cipars un viens lielais burts!</i> (angļu v. <i>Password must contain at least 8 characters, including at least one number and one capital letter!</i>).	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādas paziņojums <i>Parolei jāsaturs vismaz 8 rakstzīmes, tai skaitā vismaz viens cipars un viens lielais burts!</i> (angļu v. <i>Password must contain at least 8 characters, including at least one number and one capital letter!</i>) (skatīt 36. attēlu).
25.	Profila e-pasta sadaļā atjaunina e-pastu.	Lietotājs akurāti un precīzi aizpilda ievadlaukus, lai atjauninātu e-pastu.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādas paziņojums <i>E-pasts ir veiksmīgi atjaunināts</i> (angļu v. <i>Email updated successfully</i>).	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādas paziņojums <i>E-pasts ir veiksmīgi atjaunināts</i> (angļu v. <i>Email updated successfully</i>) (skatīt 58. attēlu).
26.	Profila e-pasta sadaļā vecais e-pasts vienāds ar jauno.	Lietotājs ievadlaukā, lai atjauninātu e-pastu ievada jauno e-pasta adresi vienādu ar veco.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādas paziņojums <i>Jaunais e-pasts ir vienāds ar veco e-pastu. Lūdzu ievadiet citu e-pastu.</i> (angļu v. <i>The new email is the same as the</i>	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādas paziņojums <i>Jaunais e-pasts ir vienāds ar veco e-pastu. Lūdzu ievadiet citu e-pastu.</i> (angļu v. <i>The new email is the same as the</i>

			<i>old email. Please enter a different email.).</i>	<i>different email.)</i> (skatīt 44. attēlu).
27.	Profila e-pasta sadaļā e-pasts jau eksistē.	Lietotājs mēģinot nomainīt e-pastu ievada jau eksistējošu e-pastu.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>E-pasts jau eksistē. Ievadiet citādāku e-pastu.</i> (angļu v. <i>This email already exists. Please enter a different email.).</i>	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>E-pasts jau eksistē. Ievadiet citādāku e-pastu.</i> (angļu v. <i>This email already exists. Please enter a different email.)</i> (skatīt 41. attēlu).
28.	Profila e-pasta sadaļā e-pasta tukši ievadlauki.	Lietotājs mainot e-pastu aizpildīt vienu vai vairākus ievadlaukus.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Lūdzu aizpildiet visus ievadlaukus.</i> (angļu v. <i>Please fill in all the fields.).</i>	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Lūdzu aizpildiet visus ievadlaukus.</i> (angļu v. <i>Please fill in all the fields.)</i> (skatīt 41. attēlu).
29.	Profila e-pasta sadaļā nepareizs e-pasts.	Lietotājs mainot e-pastu ievada nepareizu pašreizējo e-pasta adresi.	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Ievadītā e-pasta adrese neatbilst jūsu pašreizējai e-pasta adresei. Lūdzu mēģiniet vēlreiz.</i> (angļu v. <i>The email you entered does not match your current email. Please try again.).</i>	Pēc pogas <i>Atjaunināt</i> (angļu v. <i>Update</i>) nospiešanas zem tās parādās paziņojums <i>Ievadītā e-pasta adrese neatbilst jūsu pašreizējai e-pasta adresei. Lūdzu mēģiniet vēlreiz.</i> (angļu v. <i>The email you entered does not match your current email. Please try again.)</i> (skatīt 43. attēlu).
30.	Profila sadaļa dzēst kontu.	Lietotājs aiziet uz sadaļu dzēst kontu un uzspiež uz pogas <i>Dzēst kontu</i> (angļu v. <i>Delete Account</i>).	Pēc pogas uzspiešanas lietotājam pārtrauc sesiju, pārvietojot to uz pieslēgšanās vietni, un izdzēš tā datus no datu bāzes.	Pēc pogas uzspiešanas lietotājam pārtrauc sesiju, pārvietojot to uz pieslēgšanās vietni, un izdzēš tā datus no datu bāzes (skatīt 59. attēlu).
31.	Poga "Iziet" (angļu v. <i>EXIT</i>).	Lietotājs uzspiež uz pogas, kas ir novietota vietnes labā sāna augšdaļā ar nosaukumu <i>Iziet</i> (angļu v. <i>EXIT</i>).	Pēc pogas nospiešanas lietotājam pārtrauc sesiju un pārvieto to uz pieslēgšanās vietni.	Pēc pogas nospiešanas lietotājam pārtrauc sesiju un pārvieto to uz pieslēgšanās vietni (skatīt 69. attēlu).

Register

Nickname

Email

Password

Repeat Password

Register

All fields are required!
Please try again!

23. attēls. Tukši reģistrācijas ievadlauki

Register

Nickname

Email

Password

Repeat Password

Register

Password must contain at
least 8 characters, including
at least one number and one
capital letter!

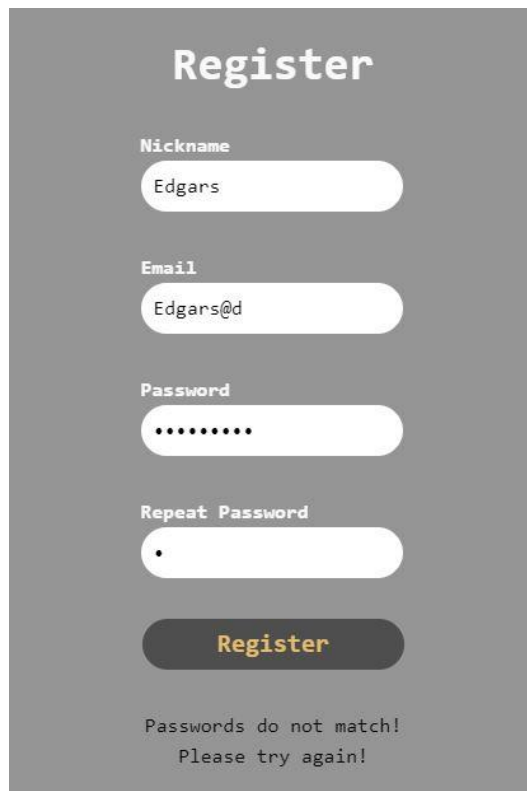
24. attēls. Nedroša lietotāja parole

A screenshot of a registration form titled "Register" on a dark gray background. The form contains four input fields: "Nickname" with the value "Edgars", "Email" with the value "Edgars@d", "Password" with masked characters ".....", and "Repeat Password" with masked characters ".....". Below the fields is a dark gray "Register" button with yellow text. At the bottom, a message reads: "Please enter a valid email address!".

25. attēls. Nepareiza e-pasta adrese

A screenshot of the same registration form titled "Register". The "Nickname" field contains "Edgars", the "Email" field contains "Edgars@d", and the "Password" field contains masked characters ".....". The "Repeat Password" field is empty and contains the placeholder text "Repeat Password". The "Register" button is visible. At the bottom, a message reads: "All fields are required! Please try again!".

26. attēls. Tukšs reģistrācijas lauks



Register

Nickname
Edgars

Email
Edgars@d

Password
.....

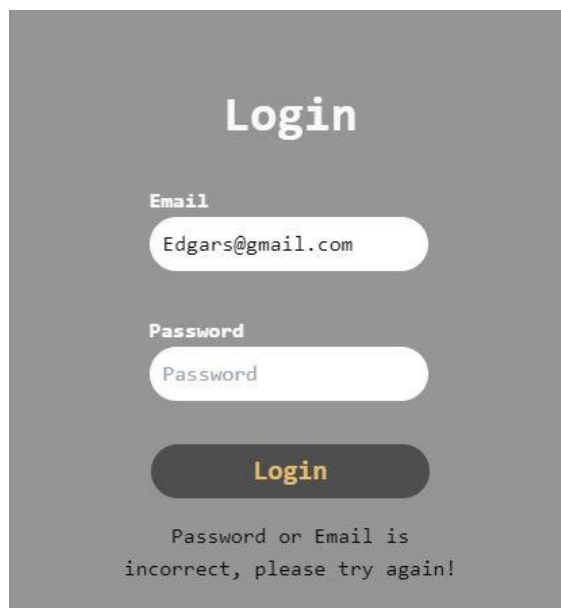
Repeat Password
.

Register

Passwords do not match!
Please try again!

This is a registration form with a grey background. It contains four input fields: 'Nickname' with the value 'Edgars', 'Email' with the value 'Edgars@d', 'Password' with masked characters '.....', and 'Repeat Password' with a single dot '.'. Below the inputs is a dark grey button with the text 'Register' in orange. At the bottom, a message states 'Passwords do not match! Please try again!'.

27. attēls. Paroles ievadlaukos nesakrīt



Login

Email
Edgars@gmail.com

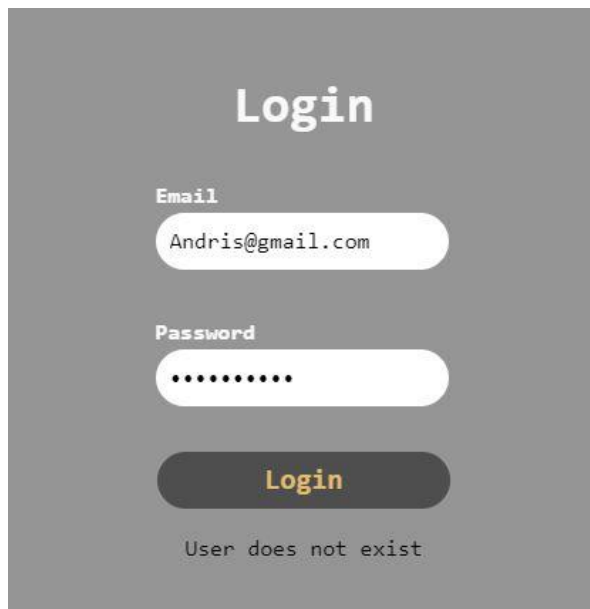
Password
Password

Login

Password or Email is
incorrect, please try again!

This is a login form with a grey background. It contains two input fields: 'Email' with the value 'Edgars@gmail.com' and 'Password' with the value 'Password'. Below the inputs is a dark grey button with the text 'Login' in orange. At the bottom, a message states 'Password or Email is incorrect, please try again!'.

28. attēls. Tukšs pieslēgšanās ievadlauks



A login form on a grey background. At the top, the word "Login" is centered in white. Below it, there are two input fields: "Email" containing "Andris@gmail.com" and "Password" containing eight dots. A dark grey "Login" button is below the password field. At the bottom, the text "User does not exist" is displayed in a small, light grey font.

29. attēls. Pieslēgšanās ar neeksistējošu lietotāju



A betting interface for a "Flyweight Bout" between two fighters. On the left is a fighter icon labeled "3 Kai Kara-France" with a checked selection box. On the right is a fighter icon labeled "7 Amir Albazi" with an unchecked selection box. In the center, it says "Flyweight Bout VS 50 ODDS 50". Below this, there are five buttons with values: "10", "20", "50", "100", and "200".

30. attēls Izvēles rūtiņa



A red-bordered message box with a black background. The text inside is "Attention! Please choose a fighter." in red.

31. attēls. Neveiksmīgas likmes paziņojums



A red-bordered message box with a black background. The text inside is "Error! You do not have enough funds to place a bet!" in red.

32. attēls. Likme bez virtuālās naudas

Account Settings

First Name	Location
<input type="text" value="Ralfs"/>	<input type="text" value="Latvia"/>
Last name	Age
<input type="text" value="Dambitis"/>	<input type="text" value="16"/>

You have to be 18 years old or older!

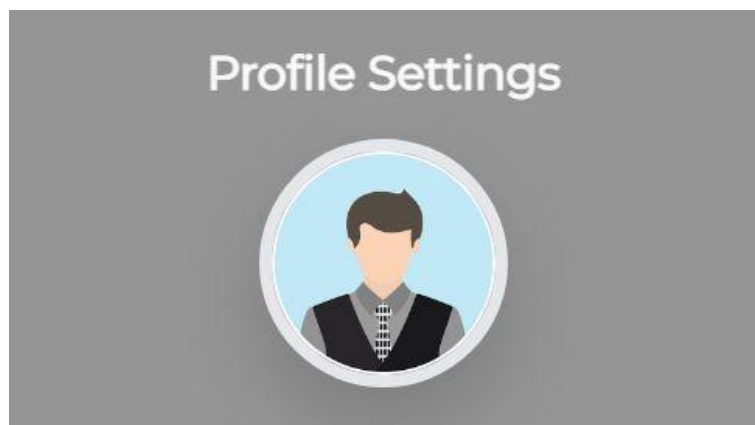
33. attēls. Lietotāja vecums

Account Settings

First Name	Location
<input type="text" value="Ralfs"/>	<input type="text" value="Latvia"/>
Last name	Age
<input type="text" value="Enter your last name"/>	<input type="text" value="20"/>

All fields must be filled!

34. attēls. Tukši konta ievadlauki



35. attēls. Pievienots konta attēls

Update password

Update

Password must contain
at least 8 characters,
including at least one
number and one capital
letter!

36. attēls. Nedroša jaunā parole

Update password

Update

Your new password is
the same as your old
password. Please choose
a different password.

37. attēls. Pašreizējā parole vienāda ar jauno

Update password

.....

.....

Repeat new password

Update

Please enter your new password.

38. attēls. Tukši paroles ievadlauki

Update password

.....

.....

.....

Update

Your current password is incorrect. Please try again.

39. attēls. Nepareiza pašreizējā parole



Update password

.....

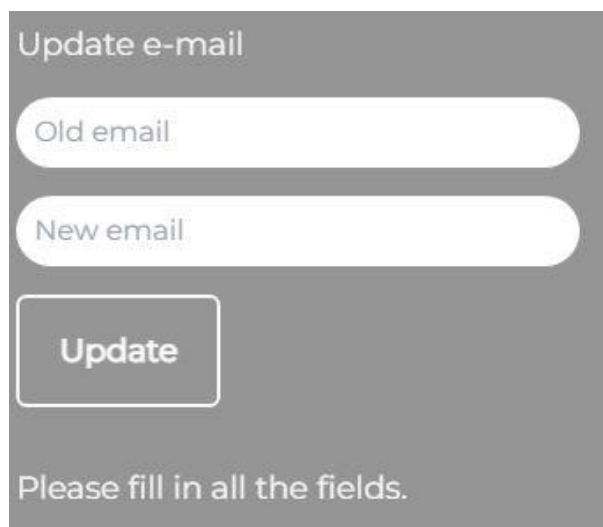
.....

.....

Update

Passwords do not match. Please try again.

40. attēls. Nesakrīt jaunās paroles



Update e-mail

Old email

New email

Update

Please fill in all the fields.

41. attēls. Tukši e-pasta adreses ievadlauki

Update e-mail

toms@gmail.com

edgars@gmail.com

Update

This email already exists.
Please enter a different email.

42. attēls. E-pasta adrese jau eksistē

Update e-mail

roberts@gmail.com

lauris@gmail.com

Update

The email you entered
does not match your
current email. Please try
again.

43. attēls. Esošais e-pasts ir nepareizs

Update e-mail

dambitis@gmail.com

dambitis@gmail.com

Update

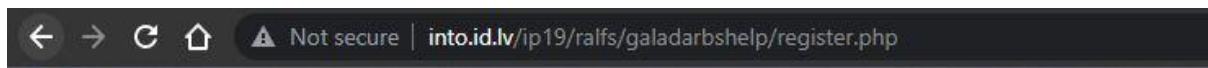
The new email is the same as the old email.
Please enter a different email.

44. attēls. Jaunais e-pasts vienāds ar esošo

9. LIETOTĀJA CEĻVEDIS

Pieslēgšanās vietnei:

Tīmekļa vietnei pieslēdzas, ievadot meklēšanas joslā vietnes saiti <http://into.id.lv/ip19/ralfs/galadarbshelp/register.php> (skatīt 45. attēlu).

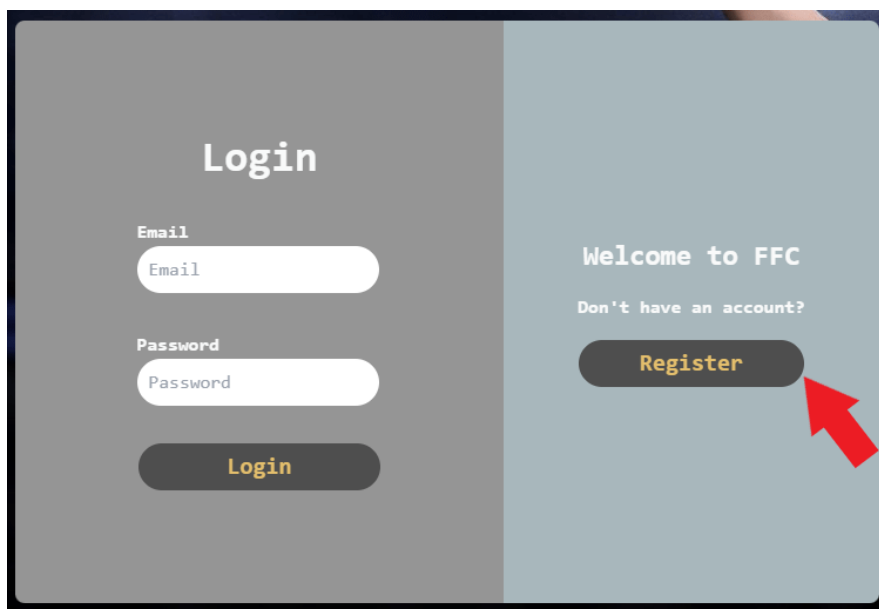


45. attēls. Tīmekļa vietnes saite

Konta izveide:

Pieslēdzoties vietnes vidū ir novietota kastīte ar informāciju. Kaste ir sadalīta divās daļās. Lai izveidotu kontu, ir jāuzspiež uz pogas ar nosaukumu *Reģistrēties* (angļu v. *Register*), kas atrodas kastes labajā malā (skatīt 46. attēlu).

Nospiežot uz pogas, lietotājam tiks parādīti ievadlauki priekš reģistrācijas. 1. ievadlaukā lietotājam ir jāievada lietotājvārds. 2. ievadlaukā lietotājam ir jāievada e-pasta adrese, kas ir derīga. Piemēram, andris@gmail.com. 3. ievadlaukā lietotājam ir jāievada sevis izdomātā parole, kurā ietilpst 8 rakstzīmes, tai skaitā vismaz viens cipars un viens lielais burts. 4. ievadlaukā lietotājam ir jāatkārto sevis izdomātā parole. Pēc ievadlauku aizpildīšanas lietotājam ir jānospiež poga *Reģistrēties* (angļu v. *Register*)(skatīt 47. attēlu). Pēc pogas nospiešanas parādīsies paziņojums zem pogas pasakot lietotājam, ka konts ir izveidots veiksmīgi.



46. attēls. Reģistrācijas poga

Register

Nickname 1.
Gatis

Email 2.
gatis@gmail.com

Password 3.
.....

Repeat Password 4.
.....

Register

The data has been added successfully!

Welcome to FFC

Have an account?

Login

47. attēls. Dati pievienoti veiksmīgi

Pieslēgšanās vietai ar izveidoto kontu:

Pēc konta izveides lietotājam ir jānospiež uz pogas *Pieslēgties* (angļu v. *Login*), kas ir novietota kastes labajā malā (skatīt 48. attēlu). Kad lietotājs ir nospiedis pogu viņu pārvietos uz pieslēgšanās ievadlaukiem. 1. ievadlauks ir paredzēts sevis pierēģistrētās e-pasta adreses ierakstīšanai. 2. ievadlauks ir paredzēts paroles ievadīšanai, kas tika izveidot priekš konta. Pēc datu savadīšanas ir jānospiež poga *Pieslēgties* (angļu v. *Login*), kas ir novietota zem ievadlaukiem (skatīt 49. attēlu).

Register

Nickname
Nickname

Email
Email

Password
Password

Repeat Password
Repeat Password

Register

Welcome to FFC

Have an account?

Login

48. attēls. Pieslēgšanās poga

Login

1.
Email
gatis@gmail.com

2.
Password
.....|

Login

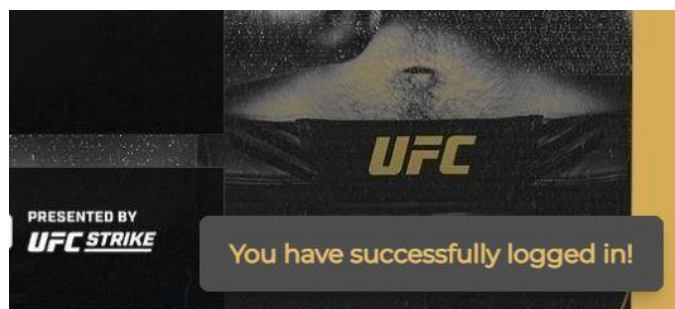
Welcome to FFC

Don't have an account?

Register

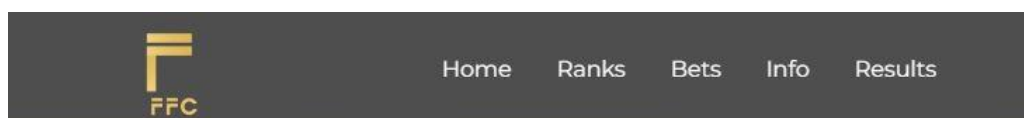
49. attēls. Pareizi aizpildīta pieslēgšanās forma

Kad poga ir nospiesta lietotājs tiek pārvietots uz tīmekļa vietnes *Sākulmapu* (angļu v. *Home*), kur tiek paskaidrota pamatinformācija par tīmekļa vietni un tās darbību. Kā arī par veiksmīgu pievienošanos parādās paziņojums labā sāna apakšdaļā, paziņojot lietotājam, ka ir veiksmīgi pieslēdzies vietnei (angļu v. *You have successfully logged in!*)(skatīt 50. attēlu).



50. attēls. Veiksmīga pieslēgšanās

Pēc pieslēgšanās lietotājam ir pieejama navigācijas joslas (skatīt 51. attēlu), kuru izmantojot lietotājs var pārvietoties pa mājaslapas tīmekļu vietnēm. Pēc jeb kuras kustības caur vietnēm, lietotājs var nokļūt atpakaļ uz tīmekļa vietnes sāukmlapu, nospiežot uz hipersaites *Sāukmlapa* (angļu v. *Home*), kas atrodas uz tīmekļa vietnes navigācijas joslas vai uz logotipa, kas ir novietots navigācijas joslas kreisajā malā. Uz navigācijas joslas lietotājam tiek parādīts lietotājvārds un virtuālās naudas kopsumma (skatīt 52. attēlu). Katrs lietotājs iesāk ar 1000 virtuālas naudas kopsummu.



51. attēls. Navigācijas josla



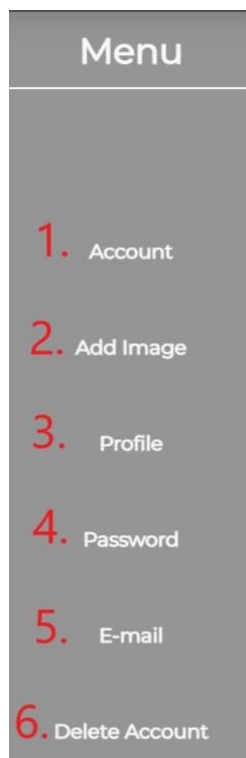
52. attēls. Lietotāja virtuālās naudas kopsumma

Konta konfigurēšana:

Pēc pieslēgšanās lietotājam ir jādodas uz profila konfigurācijas tīmekļa vietni. Uz profila vietni var nonākt uzspiežot uz pogas, kas ir attēlotā kā lietotāja ikona (skatīt 69. attēlu), tā atrodas tīmekļa vietnes navigācijas joslas labajā sānā pa kreisi no *Iziet* (angļu v. *EXIT*) hipersaites. Nospiežot uz pogas, lietotāju pārvieto uz profila tīmekļa vietni, kur lietotājs var veikt konta konfigurēšanu. Profila tīmekļa vietne ir sadalīta kastē, kreisajā malā ir navigācijas josla (skatīt 53. attēlu), kur var pārvietoties cauri lietotāja profila konfigurācijai. Labajā malā attēlo nospiešanās konfigurācijas pogas izvēlni un ievadlaukus.

Pirmā sadaļa ir *Konts* (angļu v. *Account*), kur lietotājs veic savu pirmo konfigurāciju saistība ar kontu. Konta sadaļā lietotājam ir četri ievadlauki, 1. ievadlaukā lietotājam ir jāievada savs vārds (angļu v. *First Name*). 2. ievadlaukā ir jāievada savs uzvārds (angļu v. *Last Name*). 3. ievadlaukā ir jāievada sava atrašanās vieta (angļu v. *Location*), tas ir domāts, lai varētu noteikt no

kuras valsts lietotājs nāk. 4. ievadlaukā ir jāievada savs vecums (angļu v. *Age*), jāatceras tas, ka lietotājs nedrīkst būt jaunāks par 18 gadiem, ja lietotājs ir jaunāks par 18 gadu vecumu, tad neļaus ievietot datus. Pēc datu ievadīšanas ir jānospiež poga *Pabeigt* (angļu v. *Done*), ja ievades lauki ir veiksmīgi parādīsies paziņojums zem pogas, informējot lietotāju, ka dati ir pievienoti veiksmīgi(angļu v. *Data has been added successfully!*)(skatīt 54. attēlu).



53. attēls. Profila navigācijās josla

A screenshot of the 'Account Settings' form. The form has a grey background and a white header labeled 'Account Settings'. Below the header, there are four input fields: 'First Name' (labeled 1.), 'Last name' (labeled 2.), 'Location' (labeled 3.), and 'Age' (labeled 4.). The 'First Name' field contains 'Ralfs', the 'Last name' field contains 'Dambitis', the 'Location' field contains 'Latvia', and the 'Age' field contains '20'. Below the fields, there is a 'Done' button with a red arrow pointing to it. At the bottom of the form, there is a message: 'Data has been added successfully!'.

54. attēls. Veiksmīga konta informācija

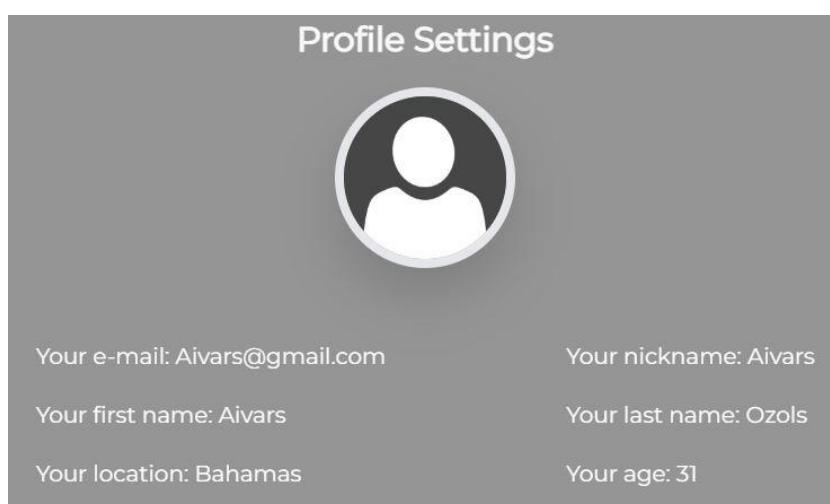
Otrā sadaļa ir *Pievienot attēlu* (angļu v. *Add Image*), kur lietotājs varēs pievienot savu attēlu priekš konta. Lai nokļūtu šajā sadaļā ir jātgriežas pie kreisās kastes malas, kur atrodas navigācijas

joslā. Uzspiežot uz pievienot attēlu sadaļa lietotājam parādīs faila ievadlauku uz kuru ir jānospiež, tad lietotājam uzleks faila izvēlne viņa datorā, kur pēc tam ir jāatzīmē izvēlētais attēls. Pēc attēla izvēlēšanās ir jānospiež poga *Pievienot* (angļu v. *Add*), kas pēc tam pievienos attēlu kontam (skatīt 55. attēlu).



55. attēls. Pievienot attēlu kontam

Trešā sadaļā ir *Profils* (angļu v. *Profile*), kur tiek attēlota lietotāja informācija. Šo sadaļu var atrast navigācijas joslā kastes kreisajā malā. Pēc navigācijas pogas nospiešanas, lietotājam kastes informācija mainās uz lietotāja konta datiem. Šajā sadaļā kastes vidū tiek attēlota lietotāja - attēls, e-pasta adrese, vārds, uzvārds, lokācija, lietotājvārds un vecums (skatīt 56. attēlu).



56. attēls. Lietotāja profila sadaļa

Ceturtā sadaļa ir *Parole* (angļu v. *Password*), kur lietotājam tiek dota iespēja mainīt paroli. Šī sadaļa atrodas uz kastes navigācijas joslās. Šai sadaļai ir trīs ievadlauki. 1. ievadlaukā lietotājam ir jāievada sava pašreizējā parole. 2. ievadlaukā ir jāievada sevis izdomātā jaunā parole. 3. ievadlaukā ir jāievada atkārtoti sevis izdomātā parole, tas ir domāts kā apstiprinājums priekš jaunās paroles. Zem ievadlaukiem atrodas poga nosaukta *Atjaunināt* (angļu v. *Update*), kad ievadlauki ir aizpildīti uz šīs pogas ir jānospiež. Ja visi lauki ir akurāti un pareizi aizpildīti zem pogas parādīsies

paziņojums, informējot lietotāju, ka viņa parole ir veiksmīgi atjaunināta (angļu v. *Your password, has been updated successfully.*)(skatīt 57. attēlu).



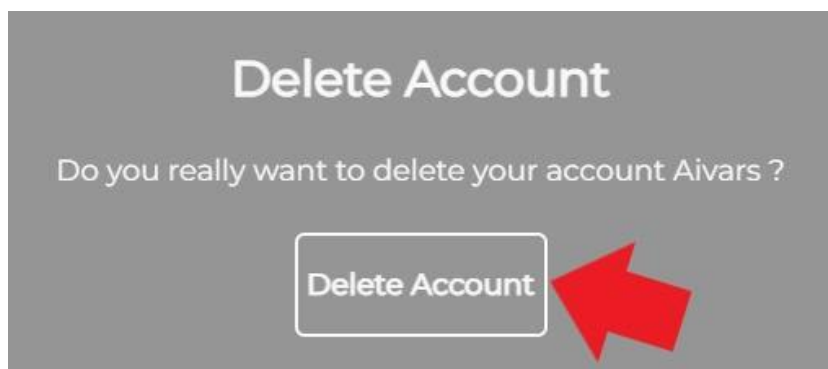
57. attēls. Veiksmīgi atjaunināta parole

Piektā sadaļa ir *E-pasts* (angļu v. *E-mail*), kur lietotājs var nomainīt e-pasta adresi gadījumā, ja viņš to ir mainījis priekš savām vajadzībām. Šī sadaļa ir novietota uz kastes navigācijas joslas. 1. ievadlaukā lietotājam ir jāievada sava pašreizējā e-pasta adrese. 2. ievadlaukā lietotājam ir jāievada sava jaunā e-pasta adrese. Kad visi šie lauki ir aizpildīti, zem tiem atrodas poga ar nosaukumu *Atjaunināt* (angļu v. *Update*) uz kuras ir jāuzspiež, lai atjauninātu datus (skatīt 58. attēls). Pēc pogas nospiešanas zem tā parādīsies paziņojums, ziņojot lietotājam, ka e-pasta adrese ir atjaunināta veiksmīgi (angļu v. *Email updated successfully*).



58. attēls. Veiksmīgi atjaunināts e-pasts

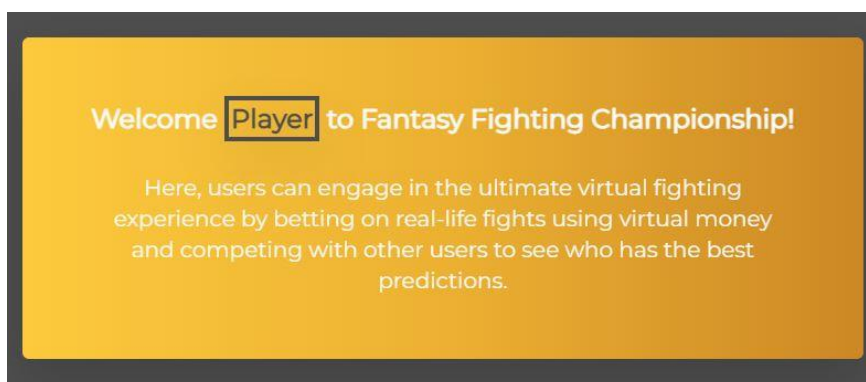
Sestā un pēdējā sadaļa ir *Dzēst kontu* (angļu v. *Delete Account*), kur lietotājs var izdzēst savu kontu vispārēji no tīmekļa vietnes un tās datu bāzes. Šī sadaļa atrodas uz kastes navigācijas joslas. Kad uzspiež uz navigācijas joslas *Dzēst kontu* kastes informācija mainās un tur tiek attēlota poga nosaukta *Dzēst kontu* (angļu v. *Delete Account*)(skatīt 59. attēlu). Vīrs pogas lietotājam pārjautā vai tiešām lietotājs vēlas dzēst kontu. Pēc pogas nospiešanas lietotāja dati tiek izdzēsti un tas tiek izmests no tīmekļa vietnes.



59. attēls. Dzēst kontu

Informācija par tīmekļa vietnes programmu:

Pēc konta konfigurācijas pabeigšanas, lietotājs var iepazīties ar spēles noteikumiem un gaitu pie tīmekļa vietnes *Informācija* (angļu v. *Info*), kas atrodas uz tīmekļa vietnes navigācijas joslas. Nospiežot uz informācijas hipersaiti lietotāju pārvieto uz informācijas tīmekļa vietni. Tur lietotājs var iepazīties ar tīmekļa vietnes mērķiem, spēles gaitu, noteikumiem un instrukciju. Tīnot zemāk tīmekļa vietnē sāks parādīties dzeltenī kvadrāti ar informāciju par spēli un tās gaitu (skatīt 60. attēlu).



60. attēls. Tīmekļa vietne informācija

Likmes veikšana:

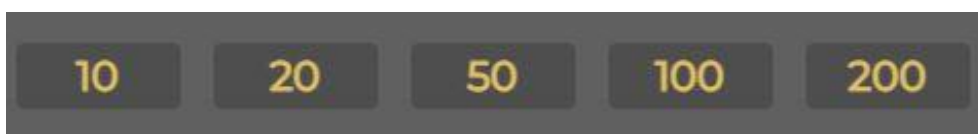
Kad lietotājs ir iepazinies ar spēles noteikumiem un tās gaitu, lietotājs var doties uz *Likmes* (angļu v. *Bets*) tīmekļa vietni, tās hipersaite atrodas uz tīmekļa vietnes navigācijas joslas. Kad lietotājs nonāk likmju vietnē tiks attēlotas tuvākā pasākuma cīņas un cīkstoņi. Tās tiks sadalītas

vairākās kastēs. Katrai cīņai sava kaste. Virs katras sekcijas ir virsraksts, kas attēlo kurā kārtī notiek šī cīņa, ja cīņa ir čempiona titulu, tad tajā kastē ir virsraksts *Čempionāts* (angļu v. *Championship*)(skatīt 61. attēlu).



61. attēls. Likmju kastes izskats

Katras cīņas kastē ir paziņota šāda informācija - cīkstoņa rangs, cīkstoņa vārds un uzvārds, cīņas svara kategorija, cīkstoņa koeficients. Kā arī katras kastes apakšā atrodas piecas pogas, kurām ir iedota vērtība, kas attēlos lietotāja likmes daudzumu. Pogu vērtības ir – 10, 20, 50, 100, 200 (skatīt 62. attēlu).



62. Attēls. Likmju pogas ar vērtībām

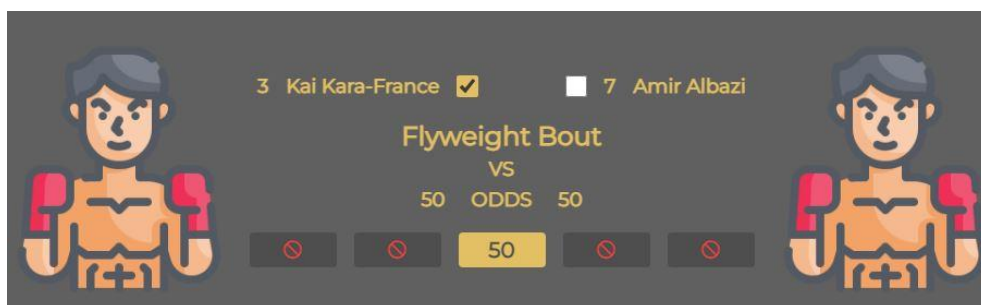
Kad visa šī informācija ir iegūta un apdomāta, lietotājs var veikt likmes par cīkstoņiem, atzīmējot izvēles rūtiņu. Katrā kastē ir divas izvēles rūtiņas, lietotājam ir jāizvēlas viena no šīm rūtiņām. Katra rūtiņa ir viens cīkstonis tai pusē kurā rūtiņa ir tuvāk cīkstoņa vārdam attēlot to specifisko cīkstoni. Kad izvēles rūtiņa ir atzīmēta var veikt likmi par izvēlēto cīkstoni (skatīt 63 attēlu).



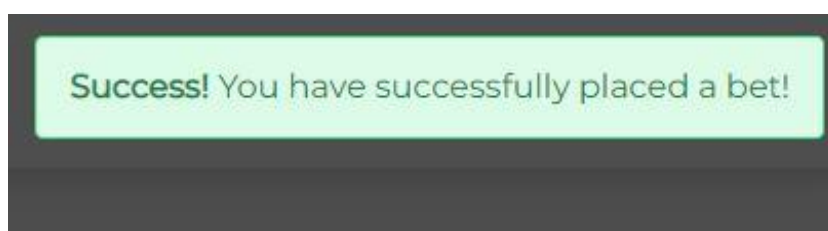
63. attēls. Cīkstoņu izvēles rūtiņas

Nospiežot vienu no piecām pogām, lietotājs veic likmi par izvēlēto cīkstoni, kur pēc tam, ja cīkstonis uzvar, tad koeficients tiek reizināts ar lietotāja izvēlēto summu. Kad izvēlēta poga tiek nospiesta atlikušās pogas nobloķējas ar sarkanu ikonu (skatīt 64. attēlu). Kā arī lietotājam vietnes labā sāna augšdaļa parādās paziņojums, ka likme ir bijusi veiksmīga (angļu v. *Success! You have successfully placed a bet!*)(skatīt 65. attēlu). Tā var turpināt likt likmes par katras cīņas vienu

cīkstoni kamēr vairs pasākumam nav cīņu vai lietotājam nebeidzas virtuālā nauda (skatīt 41. attēlu). Katru nedēļu šie dati atjaunojas ar jaunu pasākumu cīņām.



64. attēls Veiksmīga likme



65. attēls. Veiksmīgas likmes paziņojums

Likmju rezultāti:

Kad likmes ir veiktas un pasākums ir beidzies, lietotājs var doties uz *Rezultāti* (angļu v. *Results*) tīmekļa vietni. Uz šo vietni var nonākt nospiežot rezultāti hipersaiti, kas atrodas tīmekļa vietnes navigācijas joslā. Nonākot vietnē, lietotājam tiks attēlota katras uzminētās cīņas, cīkstoņa statistika un laimests. Piemēram, Cīkstoņa vārds un uzvārds, likmes daudzums, iznākums, saņemtā summa, kopējais raundu skaits, pēdējais raunds, pēdējās minūtes un cīņas (skatīt 66. attēlu) uzvaras veids. Pēc šiem datiem lietotājs var saprast cik lielu virtuālās naudas summu ir laimējis. Pēc rezultātu apskatīšanas lietotāja summa atjaunināsies un tiks papildināta (skatīt 52. attēlu).

Fighter: Albazi, Amir Bet Amount: \$100 Outcome: Won Amount Received: \$259 Fight Rounds: 5 Final Round: 5 Final Minute: 5:00 Method: Decision Split	Fighter: Miller, Jim Bet Amount: \$100 Outcome: Won Amount Received: \$337 Fight Rounds: 3 Final Round: 1 Final Minute: 0:23 Method: Ko Tko	Fighter: Caceres, Alex Bet Amount: \$100 Outcome: Won Amount Received: \$310 Fight Rounds: 3 Final Round: 3 Final Minute: 5:00 Method: Decision Unanimous
--	---	---

66. attēls. Likmju un cīņu rezultāti

Ranga noskaidrošana:

Pēc likmju rezultātu pārbaudīšanas lietotājs var doties uz navigācijas joslā novietoto hipersaiti, kas lietotāju nogādās uz *Rangi* (angļu v. *Ranks*) tīmekļa vietni. Šajā vietnē tiek attēloti

tīmekļa vietnes labākie piecdesmit spēlētāji (skatīt 68. attēlu), trīs no tiem novietoti pašā augšā (skatīt 67. attēlu). Ja lietotājs ir viens no piecdesmit, tad varēs redzēt kurā vietā ir nonācis pēc sevis veiktajām likmēm.



67. attēls. Trīs labākie lietotāji

Top 50 Players in Fantasy Fighting Championship			
Rank	4	Olegs	1000
Rank	5	Artis	1000
Rank	6	Arturs	1000
Rank	7	Gatis	1000
Rank	8	RalfsD	989
Rank	9	Ainars	980
Rank	10	Marcis	950

68. attēls. Piecdesmit labākie lietotāji

Atvienošanās no tīmekļa vietnes:

Kad visas darbības ir veiktas un lietotājs vēlas pārtraukt sesiju uz navigācijas joslas ir novietota hipersaite *Iziet* (angļu v. *EXIT*) (skatīt 69. attēlu). Nospiežot uz šīs hipersaites, lietotāja sesija tiks pārtraukta, un tas tiks pārvietots uz pieslēgšanās lapu (skatīt 29. attēlu).



69. attēls. Atslēgties no konta

SECINĀJUMI UN PRIEKŠLIKUMI

1. Darba mērķis tika sasniegts – kvalifikācijas darbs ir izstrādāts ar jautru un konkurētspējīgu tīmekļa vietni priekš draugiem un *Mixed Martial Arts* faniem, kas rosina sāncensību starp vienu un otru paralēli padarot to jautru.
2. Tīmekļa vietnei tika izveidots pievilcīgs dizains ar vienādu krāsu paleti, padarot to viegli salasāmu un saprotamu priekš ikviena lietotāja.
3. Priekšgala sistēma (angļu v. *front-end*) likmēm tika veiksmīgi izstrādāta. Veiksmīgi izveidotas pogas ar likmēm un izvēles rūtiņas, lai lietotājs varētu veikt likmes uz specifisku lietotāju noteiktā cīņā un pasākumā. Katrā laukā lietotājs pirms veic likmi redz cīkstoņa koeficientu, svara kategoriju un rangū, kas palīdz veikt izglītotu likmi uz cīkstoni. Pēc likmes veikšanas lietotājam noņem summu no virtuālās valūtas kopsummas. Kad pasākums ir beidzies lietotājs saņem noteikto virtuālās valūtas summu par veiksmīgi uzminētu cīņu.
4. Veiksmīgi tika izveidota datu bāzes arhitektūra ar ER diagrammas palīdzību. Datu bāze ir strukturēta un savienojas savstarpēji ar relācijām un apvieno datus kopā. Veiksmīgi padod un ievieto datus, kad tie ir nepieciešami.
5. Aizmugures sistēma (angļu v. *back-end*) tika veiksmīgi īstenota. Tā glabā datus par lietotāju, cīkstoņiem, pasākumiem, likmēm un rezultātiem. Dati par cīņām tika īstenoti ar API palīdzību, padodot tos uz datu bāzes tabulām, un glabājot tos tur. Informācija par lietotāja likmēm tiek veiksmīgi saglabāta ar visu detalizēto informāciju par kuru cīņu un cīkstoni, lietotājs ir veicis likmi. Kā arī pēc noteiktā laika par veiksmīgu likmi, lietotājam tiek iedota virtuālā nauda, kad rezultāti tiek atjaunināti balstoties pēc lietotāja likmes.
6. Tīmekļa vietnei tika izveidoti testi un pārbaudes, lai tā spētu darboties pareizi un bez kļūdām. Vietnes ievadlauki izvada kļūdu ziņojumus, ja lietotājs tos nav pareizi aizpildījis. Kā arī, ja lietotājs veic neveiksmīgas likmes ar noteiktām kļūdu pazīmēm, tad tiek izvadīts kļūdas paziņojums. Katrs ievades lauks un pogas tika veiksmīgi pārbaudītas ar katru kļūmes pazīmi, lai tie izvadītu kļūdu paziņojumus.
7. Gandrīz katra radusies problēma tika novērsta ar korektu un precīzu pieeju, meklējot problēmas cēloni un izcelsmi.
8. Tīmekļa vietnē plānoju veikt uzlabojumus un optimizēt to ar lielāku funkcionalitāti.
9. Plānoju pārveidot savu programmu objekt orientētu, lai tā būtu vieglāk salasāma, funkcionālāka.
10. Pievienot likmju (angļu v. *Bets*) tīmekļa vietnes sadaļai lietotāja paša ierakstītu likmes ievadlauku, noņemot pogas.

11. Pievienot detalizētāku aprakstu par katru cīkstoni un pievienot statistiku tam.
12. Pārveidot rezultātu tīmekļa vietnes sadaļu, uztaisot to vieglāk salasāmu un pārredzamāku.
Kā arī pievienot tai precīzāku statistiku, kas paralēli plašāk papildinās to.
13. Pie lietotāja profila (angļu v. *Profile*) sadaļas tīmekļa vietnē pievienot ranga lomu.

IZMANTOTĀ LITERATŪRA

1. Bloom, Joe. *How Popular Is MMA as Compared to Soccer and Basketball?* [tiešsaiste]. MMAHIVE, 2022 [skatīts 19.12.2022]. Pieejams: <https://www.mmahive.com/ufc-viewership-statistics/>
2. Marconi, Kenneth. *UFC Viewership Statistics: PPV Buys & Attendance* [tiešsaiste]. Victory Fighter, 2022 [skatīts 19.12.2022]. Pieejams: <https://victoryfighter.com/how-popular-is-mma-as-compared-to-soccer-and-basketball/>
3. Functional and Nonfunctional Requirements: Specification and Types [tiešsaiste]. Altexsoft, 2021 [skatīts 19.12.2022]. Pieejams: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/#:~:text=Functional%20requirements%20are%20product%20features,system%20behavior%20under%20specific%20conditions.>
4. O'Grady, Brian. *What is HTML? An Introduction* [tiešsaiste]. Code Institute, 2022 [skatīts 19.12.2022]. Pieejams: <https://codeinstitute.net/global/blog/what-is-html-and-why-should-i-learn-it/>
5. Morris, Scott. *Tech 101: The Ultimate Guide To CSS* [tiešsaiste]. Skillcrush, 2022 [skatīts 19.12.2022]. Pieejams: <https://skillcrush.com/blog/css/>
6. Fitzgerald, Anna. *Tailwind CSS: What It Is, Why Use It & Examples* [tiešsaiste]. Hubspot, 2022 [skatīts 19.12.2022]. Pieejams: <https://blog.hubspot.com/website/what-is-tailwind-css>
7. Kolade, Chris. *What is PHP? The PHP Programming Language Meaning Explained* [tiešsaiste]. Freecodecamp, 2021 [skatīts 19.12.2022]. Pieejams: <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/>
8. Krüger, Gerhard un Lane, Charles. *How to Write a Software Requirements Specification (SRS Document)* [tiešsaiste]. Perforce, 2023 [skatīts 25.05.2023]. Pieejams: <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>
9. *Functional Requirements Specification (FRS)* [tiešsaiste]. Presentationeze, 2013 [skatīts 25.05.2023]. Pieejams: [https://www.presentationeze.com/presentations/software-validation/software-validation-full-details/functional-requirements-specification-frs/#:~:text=The%20functional%20requirements%20specification%20is,%E2%80%93%20User%20Requirements%20Specification\)%20document](https://www.presentationeze.com/presentations/software-validation/software-validation-full-details/functional-requirements-specification-frs/#:~:text=The%20functional%20requirements%20specification%20is,%E2%80%93%20User%20Requirements%20Specification)%20document)
10. Moen, Alexander. *Defining Requirements and Specifications* [tiešsaiste]. Argondigital, 2022 [skatīts 30.05.2023]. Pieejams: <https://argondigital.com/blog/product-management/requirements-vs-specifications-create-a-shared-vocabulary/>
11. Paruch, Zach. *What Is JavaScript & What Is It Used For? A Basic Guide to JS* [tiešsaiste]. Semrush Blog, 2023 [skatīts 08.06.2023]. Pieejams: <https://www.semrush.com/blog/javascript/>
12. B, Richard. *What is MySQL: MySQL Explained For Beginners* [tiešsaiste]. Hostinger, 2023 [skatīts 08.06.2023]. Pieejams: <https://www.hostinger.com/tutorials/what-is-mysql>

13. *CronJob* [tiešsaiste]. Kubernetes, 2023 [skatīts 08.06.2023]. Pieejams: <https://kubernetes.io/docs/concepts/workloads/controllers/cron-jobs/>
14. *How does software testing work?* [tiešsaiste]. IBM, 2023 [skatīts 08.06.2023]. Pieejams: <https://www.ibm.com/topics/software-testing>
15. *OOAD - Functional Modeling* [tiešsaiste]. Tutorialspoint, 2023 [skatīts 08.06.2023]. Pieejams: https://www.tutorialspoint.com/object_oriented_analysis_design/ood_functional_modeling.htm
16. Vats, Rohan. *Foreach Loop In PHP: Definition, Functions & Uses* [tiešsaiste]. Upgrad, 2020[skatīts 09.06.2023]. Pieejams: <https://www.upgrad.com/blog/foreach-loop-in-php/>
17. Rouse, Margaret. *What is an Integer (INT)? - Definition from Techopedia* [tiešsaiste]. Techopedia, 2016 [skatīts 09.06.2023]. Pieejams: <https://www.techopedia.com/definition/832/integer-int>
18. *What is an if Statement?* [tiešsaiste]. Computer Hope, 2019 [skatīts 09.06.2023]. Pieejams: <https://www.computerhope.com/jargon/i/ifstatme.htm>
19. *What is Array?* [tiešsaiste]. Geeksforgeeks, 2023 [skatīts 09.06.2023]. Pieejams: <https://www.geeksforgeeks.org/what-is-array/>
20. Busbee, Leroy Kenneth un Braunschweig, Dave. *Strings – Programming Fundamentals* [tiešsaiste]. Rebus community, 2018 [skatīts 09.06.2023]. Pieejams: <https://press.rebus.community/programmingfundamentals/chapter/strings/>
21. *Float* [tiešsaiste]. Computer Science Wiki, 2019[skatīts 09.06.2023]. Pieejams: <https://computersciencewiki.org/index.php/Float>
22. *PHP: while – Manual* [tiešsaiste]. Php, 2023 [skatīts 09.06.2023]. Pieejams: <https://www.php.net/manual/en/control-structures.while.php>
23. *Creating, Destroying, and Working With Session in PHP* [tiešsaiste]. Simplilearn, 2023 [skatīts 09.06.2023]. Pieejams: <https://www.simplilearn.com/tutorials/php-tutorial/session-in-php>

PIELIKUMI

1. pielikums. Lietotāja reģistrācijas programmas kods

Lai lietotājam būtu piekļuve pie tīmekļa vietnes un spētu veikt likmes uz cīņām, tam vispirms ir jāveic reģistrācija (angļu v. *Registration*). Reģistrācija lietotājam sniedz iespēju pēc tam pieslēgties vietnei, kas ļaus veikt likmes un daudz citu darbību pašā tīmekļa vietnē. Reģistrācijai kopā ir 4 ievadlauki, kur lietotājam visi ir jāaizpilda korekti un precīzi, lai veiktu reģistrāciju.

Pirmajā ievadlaukā lietotājam ir jāievada derīga e-pasta adrese. E-pasta adreses lauks izmanto simbolu virknes mainīgo (angļu. *string*) datu tipu, lai glabātu informāciju datu bāzē.

Otrais ievadlauks ir lietotāja lietotājevārda ievadīšana. Lai saglabātu šī ievadlauka ievadīto datu bāzē tiek izmantots simbolu virknes mainīgo (angļu. *string*) datu tips.

Trešais ievadlauks ir lietotāja paroles ievadīšana, kur ir nepieciešams ievadīt vismaz 8 rakstzīmes ar vismaz vienu lielo burtu un 1 ciparu. Lai saglabātu šī ievadlauka ievadīto datu bāzē tiek izmantots simbolu virknes mainīgo (angļu. *string*) datu tips.

Ceturtais ievadlauks ir lietotāja paroles atkārtošana, kurai nepieciešams būt identiskai ar trešā ievadlauka ievadīto. Lai saglabātu šī ievadlauka ievadīto datu bāzē tiek izmantots simbolu virknes mainīgo (angļu. *string*) datu tips.

Priekš ievadlaukiem un to izvietojumam izmantoju *html* un priekš stila *tailwindcss*.

```
<!-- Ievades forma reģistrācijas-->
<form class="forms flex flex-col" id="form">
  <div id="divReg" class="hidden flex-col justify-center items-center gap-8">
    <h1 class="text-4xl font-bold text-[#F9FAFA]">Register</h1>
    <div>
      <!-- Lietotājevārda ievades lauks priekš reģistrācijas-->
      <p class="text-1xl font-semibold text-[#F9FAFA]">Nickname</p>
      <input class="rounded-full w-[99%] h-[2.5rem] outline-none p-2.5" id="nickname"
type="text"
      name="nickname" placeholder="Nickname">
    </div>
    <div>
      <!-- E-pasta ievades lauks priekš reģistrācijas-->
      <p class="text-1xl font-semibold text-[#F9FAFA]">Email</p>
      <input class="rounded-full w-[99%] h-[2.5rem] outline-none p-2.5" id="email"
type="email"
      name="email" placeholder="Email">
    </div>
    <div>
      <!-- Paroles ievades lauks priekš reģistrācijas-->
      <p class="text-1xl font-semibold text-[#F9FAFA]">Password</p>
      <input class="rounded-full w-[99%] h-[2.5rem] outline-none p-2.5" id="password"
type="password"
```

```

        name="password" placeholder="Password">
    </div>
    <div>
        <!-- Atkārtotas paroles ievades lauks priekš reģistrācijas-->
        <p class="text-1xl font-semibold text-[#F9FAFA]">Repeat Password</p>
        <input class="rounded-full w-[99%] h-[2.5rem] outline-none p-2.5" id="repeatpassword"
            type="password" name="repeatpassword" placeholder="Repeat Password">
    </div>
    <!-- Ievades lauku ievietošanas poga ar javascript funkciju priekš reģistrācijas-->
    <button class="rounded-full bg-[#4E4E4E] text-[#e4c065] font-bold w-[99%] h-[2.5rem]
        text-xl hover:bg-[#e4c065] hover:text-[#4E4E4E] hover:scale-110 transition duration-150 ease-
        out hover:ease-in hover:transition duration-300 ease-out"
        onclick="getInput('registration.php',event, 'form')">Register
    </button>
</div>
</form>

```

Priekš nepārtrauktas datu sūtīšanas izmantoju – *Javascript Ajax*. Kā arī tas tiek izmantots, lai vietne netiktu pārlādēta pēc datu nosūtīšanas uz datu bāzi.

// Funkcijas, kas ir paredzēta, lai iegūtu php failus, kas ir paredzēti priekš reģistrācijas un pieslēgšanās formām

```

function getInput(inputCtrl, event, inputForm) {
    // Novērš notikuma noklusējuma uzvedību.
    event.preventDefault();
    // Iegūst msg id no faila, kurā ir ievietots paziņojums par kļūdu.
    let msg = document.getElementById('msg');
    // Iegūst formu no faila, kurā ir ievietota forma.
    let form = document.getElementById(inputForm);
    let xmlhttp = new XMLHttpRequest();
    // Iegūst datus no formas.
    let formData = new FormData(form);
    xmlhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            // Pārbauda, vai atgrieztā vērtība ir "true".
            if (this.responseText === "true") {
                const successMessage = "You have successfully logged in!";
                // Ja ir "true", tad pārslēdz uz index.php un ievieto paziņojumu.
                window.location = `index.php?message=${encodeURIComponent(successMessage)}`;
            } else {
                msg.innerHTML = this.responseText;
            }
        }
    };
    xmlhttp.open("POST", "controllers/" + inputCtrl, true);
    xmlhttp.send(formData);
}

```

Priekš datu ievadišanas datu bāzē izmantoju MySQL un php. Katram ievadlaukam tiek izmantots php, lai ievietotu padotos datu uz datu bāzi.

```
// Dezinficē ievadītos datus no lietotāja puses un saglabā mainīgajos
$email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
$nickname = filter_var($_POST['nickname'], FILTER_SANITIZE_STRING);
$password = filter_var($_POST['password'], FILTER_SANITIZE_STRING);
$repeatpassword = filter_var($_POST['repeatpassword'], FILTER_SANITIZE_STRING);
// Pārbauda vai visi lauki ir aizpildīti
if (empty($email) || empty($nickname) || empty($password) || empty($repeatpassword)) {
    echo "All fields are required! Please try again!";
} // Pārbauda vai paroles atkārtojums sakrīt
elseif ($password !== $repeatpassword) {
    echo "Passwords do not match! Please try again!";
} // Pārbauda vai e-pasts ir derīgs
elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo "Please enter a valid email address!";
} // Pārbauda vai parole ir droša
elseif (strlen($password) < 8 || !preg_match("#[0-9]+#", $password) || !preg_match("#[A-Z]+#", $password) || !preg_match("#[a-z]+#", $password)) {
    echo "Password must contain at least 8 characters, including at least one number and one capital letter!";
} else {
    // Aizsargā pret SQL injekcijām
    $stmt = $conn->prepare("SELECT nickname FROM loginhelp WHERE nickname = ? OR email = ?");
    $stmt->bind_param("ss", $nickname, $email);
    $stmt->execute();
    $result = $stmt->get_result();
    $row = $result->fetch_assoc();
    // Pārbauda vai lietotājs jau eksistē
    if (!empty($row)) {
        echo "User already exists!";
    } else {
        // Ja lietotājs neeksistē, saglabā datus datubāzē un šifrē paroli
        $hashed_password = password_hash($password, PASSWORD_DEFAULT);
        $stmt = $conn->prepare("INSERT INTO loginhelp (email, nickname, `password`) VALUES (?, ?, ?)");
        $stmt->bind_param("sss", $email, $nickname, $hashed_password);
        // Pārbauda vai izpildījās un izvada atbilstošu paziņojumu
        if ($stmt->execute()) {
            echo "The data has been added successfully!";
        } else {
            echo "Error! Failed to add data!";
        }
    }
}
}
```

2. *pielikums*. Lietotāja pieslēgšanās programmas kods

Lai lietotājs varētu pieslēgties (angļu v. *Login*) tīmekļa vietnei ir jāaizpilda 2 ievadlauki. Šie ievadlauki ir jāaizpilda ar reģistrācijas ievadīto informāciju. Pēc pieslēgšanās lietotāju pārvieto uz sākumlapu, kur pēc tam ir pieejams visas pārējās tīmekļa vietnes funkcijas. Lietotāja pieslēgšanās tiek izmantots tas pats kods *javascript* kā reģistrācijai.

Pirmais ievadlauks ir paredzēts priekš lietotāja reģistrētās e-pasta adreses ievadīšanas. Otrais ievadlauks ir paredzēts priekš lietotāja izveidotās paroles ievadīšanas. Abi ievadlauki izmanto simbolu virknes mainīgo (angļu. *string*) datu tipus, lai ievadītu datus datu bāzē.

Priekš ievadlaukiem un to izvietojumam izmantoju *html* un priekš stila *tailwindcss*.

```
<!--      Ievades forma priekš pieslēgšanās-->
<form id="log">
<div id="divLog" class="flex flex-col justify-center items-center gap-8 w-full">
<h1 class="text-4xl font-bold text-[#F9FAFA]">Login</h1>
<div>
<!--      E-pasta ievades lauks priekš pieslēgšanās-->
<p class="text-1xl font-semibold text-[#F9FAFA]">Email</p>
<input class="rounded-full w-[99%] h-[2.5rem] outline-none p-2.5" id="email"
type="email"
      name="email2" placeholder="Email">
</div>
<div>
<!--      Paroles ievades lauks priekš pieslēgšanās-->
<p class="text-1xl font-semibold text-[#F9FAFA]">Password</p>
<input class="rounded-full w-[99%] h-[2.5rem] outline-none p-2.5" id="password"
type="password"
      name="password2" placeholder="Password">
</div>
<!--      Ievades lauku ievietošanas poga ar javascript funkciju priekš pieslēgšanās-->
<button class="rounded-full bg-[#4E4E4E] text-[#e4c065] font-bold w-[99%] h-[2.5rem]
text-xl hover:bg-[#e4c065] hover:scale-110 hover:text-[#4E4E4E] hover:transition duration-300
ease-out"
      onclick="getInput('loginVerify.php', event, 'log')">Login
</button>
</div>
</form>
```

Priekš datu pārbaudīšanas no datu bāzēs es izmantoju MySQL un php.

```
//Pārbauda vai lietotājs ir ievadījis e-pasta adresi un paroli
if (isset($_POST['email2']) && isset($_POST['password2'])) {
    $email = mysqli_real_escape_string($conn, $_POST['email2']);
    $password = mysqli_real_escape_string($conn, $_POST['password2']);
    //Iegūst datus no datu bāzes
    $stmt = $conn->prepare("SELECT * FROM loginhelp WHERE email = ?");
```

```

//Novērš SQL injekcijas
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();
//Pārbauda vai lietotājs eksistē
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        //Pārbauda vai parole un e-pasta adrese sakrīt ar datu bāzes vērtībām
        if ($email == $row['email'] && password_verify($password, $row['password'])) {
            $_SESSION['user'] = $row['id'];
            $_SESSION['logged'] = true;
            $users = "true";
        } else {
            $users = "Password or Email is incorrect, please try again!";
        }
    }
} else {
    $users = "User does not exist";
}
//Pārbauda vai visi lauki ir aizpildīti
} else {
    $users = "Fill in all fields!";
}
if (!empty($users)) {
    echo htmlentities($users, ENT_QUOTES);
}

```


3. pielikums. Skata mainīšanas programmas kods

Es izveidoju kodu, kas maina skatus starp reģistrāciju un pieslēgšanos priekš lietotāja ievadlauku ievadīšanas. Šis atvieglo lietotāja pārvietošanos tīmekļa vietnē nepieprasot vietnes atjaunināšanu, lai pārvietotos pār ievadlaukiem.

Priekš pogas un tās izvietojumu izmantoju *html* un priekš stila un dizaina izmantoju *tailwindcss*.

```
<div class="boxRight flex flex-col justify-center items-center min-h-[50vh] flex-1 bg-  
[#A8B7BC] p-6 min-w-[20rem] rounded-r-lg">  
  <div class="logRegButton flex flex-col justify-center items-center gap-4">  
    <h2 class="text-2xl font-bold text-[#F9FAFA]">Welcome to FFC</h2>  
    <h3 class="text-1xl font-bold text-[#F9FAFA]" id="text">Don't have an account?</h3>  
    <!--      Poga, kas maina skatu starp reģistrāciju un pieslēgšanos-->  
    <button class="rounded-full bg-[#4E4E4E] text-[#e4c065] font-bold w-[99%] h-[2.5rem]  
text-xl hover:bg-[#e4c065] hover:text-[#4E4E4E] hover:transition duration-300 ease-out  
hover:scale-110"  
      id="btn" onclick="displayShow()">Register  
    </button>  
  </div>  
</div>
```

Lai šis process notiktu bez vietnes atjaunināšanas es izmantoju *Javascript AJAX*. Uzspiežot uz pogas tiks mainīti ievadlauki.

// Funkcija, kas ļauj lietotājam pārvietoties no reģistrācijas uz pieslēgšanās lapu un otrādi.

```
function displayShow() {
```

// Novērš notikuma noklusējuma uzvedību.

```
event.preventDefault();
```

// Meklē divReg no faila, kurā ir ievietota reģistrācijas forma.

```
let x = document.getElementById("divReg");
```

// Pievieno attiecīgo stilu un tekstu.

```
if (x.style.display === "none") {
```

```
  btn.textContent = 'Login';
```

```
  text.textContent = "Don't have an account?";
```

```
  x.style.display = "flex";
```

```
} else {
```

```
  btn.textContent = 'Register';
```

```
  text.textContent = "Have an account?";
```

```
  x.style.display = "none";
```

```
}
```

// Meklē divLog no faila, kurā ir ievietota pieslēgšanās forma.

```
let y = document.getElementById("divLog");
```

// Pievieno attiecīgo stilu un tekstu.

```
if (y.style.display === "flex") {
```

```
  btn.textContent = 'Login';
```

```
    text.textContent = "Have an account?";  
    y.style.display = "none";  
  } else {  
    btn.textContent = 'Register';  
    text.textContent = "Don't have an account?";  
    y.style.display = "flex";  
  }  
}
```

4. pielikums. Lietotāja sesijas programmas kods

Lai lietotājs varētu pieslēgties tīmekļa vietnei tiek uzsākta sesija. Šī sesija ir tieši unikāla tam lietotājam. Ar sesiju palīdzību tiek izvadīti dati lietotāja tīmekļa vietnes iestatījumu sadaļā. Kā arī sesijas atbilst par lietotāja darbību glabāšanu, kas notiek vietnē. Lietotāja dati no datu bāzes tiek izmantoti, lai uzsāktu šo unikālo sesiju. Kad lietotājs veic likmi tā unikālais identifikators tiek pievienots klāt šai likmei, lai spētu noteikt kurš ir veicis šo likmi un kuram ir jāiedod virtuālā valūta, ka likme ir bijusi rezultatīva. Lietotājs arī var pārtraukt sesiju, atvienojoties no tīmekļa vietnes.

Lai sesiju uzsāktu es izmantoju `session_start()` un priekš konfigurācijas koda *php*.

// šī funkcija pārbauda vai sesija ir sākusies, ja nav, tad tiek pāradresēts uz reģistrācijas vietni
function **userID()**

```
{  
    // Pārbauda vai sesija ir sākusies  
    if (isset($_SESSION['logged'])) {  
        $userID = $_SESSION['user'];  
        return $_SESSION['user'];  
        // Ja nav, tad tiek pāradresēts uz reģistrācijas vietni  
    } else {  
        header("Location:./register.php");  
    }  
}
```

// Ja ir nospiesta hipersaite "Iziet", tad tiek izsaukta šī funkcija, kas iznīcina sesiju un pāradresē uz reģistrācijas vietni

function **logOut()**

```
{  
    // Pārbauda vai ir nospiesta hipersaite "EXIT", ja ir tad iznīcina sesiju un pāradresē uz reģistrācijas vietni  
    if (isset($_POST['logOut'])) {  
        session_destroy();  
        header("Location:./register.php");  
    }  
}
```

// Šī funkcija pārbauda vai lietotājs ir pieslēdzies un ja ir tad tiek pāradresēts uz index.php vietni
function **loggedIn()**

```
{  
    // Pārbauda vai sesija ir sākusies, ja ir tad tiek pāradresēts uz index.php vietni  
    if (isset($_SESSION['logged'])) {  
        header("Location:./index.php");  
    }  
}
```

5. pielikums. Lietotāja dzēšanas no datu bāzes porgrmmas kods

Ja lietotājs ir izlēmis, ka vairs nevēlas lietot tīmekļa vietni, tas var dzēst savus datus no datu bāzes. Es izveidoju kodu, kas izdzēs lietotāja datus un pārvieto uz pieslēgšanās vietni, kad dati ir izdzēsti.

Priekš ievadlaukiem un to izvietojumam izmantoju *html* un priekš stila un dizaina izmantoju *tailwindcss*.

```
<!-- Dzēst lietotāja no datu bāzes-->
<div id="deleteItems" style="display:none"
  class="deleteItems flex flex-col tabShow min-h-[91.8vh] justify-center items-center gap-6 w-[100%]">
  <div class="topTitle flex flex-col justify-center items-center">
    <h1 class="text-2xl min-[425px]:text-3xl font-bold text-[#F9FAFA]">Delete
Account</h1>
  </div>
  <!-- Attēlo lietotāja lietotārvārdu un pārvaicā vai tiešām vēlas dzēst savu kontu-->
  <div class="deleteAccount flex flex-col gap-2 justify-center items-center xl:px-52 lg:px-32 md:px-16">
    <p class="text-lg text-[#F9FAFA] text-center">Do you really want to delete your
account <?php nickNameProfile($conn); ?> ?</p>
  </div>
  <!-- Dzēst kontu poga-->
  <div class="buttonDelete flex flex-col gap-2 justify-center items-center">
    <button onclick="deleteUser(<?php echo $userID; ?>)" name="delete" id="deleteUser"
      class="rounded-md text-lg bg-[#959595] hover:bg-[#e4c065] border-2 border-[#F9FAFA] text-[#F9FAFA] w-[16vh] h-[4rem] font-bold">
      Delete Account
    </button>
  </div>
</div>
```

Es izmantoju *MySQL*, lai dzēstu lietotāja datus no datu bāzes. Lai konfigurētu visu es izmantoju *php* savā kodā.

```
$userID = $_SESSION['user'];
// Dzēs lietotāju no datubāzes
$stmt = $conn->prepare("DELETE FROM loginhelp WHERE id=?");
// Aizsargā pret SQL injekcijām
$stmt->bind_param("i", $userID);
// Pārbauda vai izpildījās
if ($stmt->execute()) {
  // Ja funkcija izpildas, tad lietotājam pārtrauc sesiju un iznīcina to.
  unset($_SESSION['user']);
  unset($_SESSION['logged']);
  session_destroy();
}
```

```

    $insertDel = 0;
} else {
    // Ja neizpildījās izvada kļūdu
    $insertDel = "Error! Please try again!";
}
echo $insertDel;

```

Lai lietotāju automātiski pēc pogas nospiešanas pārvietotu uz pieslēgšanās vietni es izmantoju *Javascript AJAX*. Kā arī tas tiek izmantots, lai viss šis process notiktu bez tīmekļa vietnes pārlādes.

// Funkcija, kas izdzēš lietotāju no datubāzes un parvieto to uz reģistrācijas lapu.

```

function deleteUser(userID) {
    let msg = document.getElementById('deleteUser');
    let xmlhttp = new XMLHttpRequest();
    // Iegūst datus no formas.
    xmlhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            if (this.responseText == 0) {
                // Ja ir "true", tad lietotāju izdzēš un pārslēdz uz reģistrācijas lapu.
                window.location = "register.php";
            } else {
                msg.innerHTML = this.responseText;
            }
        }
    };
    xmlhttp.open("GET", "controllers/deleteUser.php?" + userID, true);
    xmlhttp.send();
}

```

6. pielikums. Likmju ievietošanas programmas kods

Es izveidoju kodu, lai ievietotu lietotāja likmi datu bāzē. Kad lietotājs ir izvēlējis cīņu un cīkstoni par kuru veiks likmi, tam ir jāizvēlas izvēles rūtiņas un likmes summa. Katram likmju laukam ir piešķirta informācija par cīņu un likmes iespējas.

Izmantoju php, lai attēlotu datus no datu bāzes, kā arī izmantoju to, lai ieliktu lietotāja veikto likmi datu bāzē. Pie attēlotā izmantoju *html* un *tailwindcss*.

```
//Izvēlas datus no tabulas UFC_Single_Event 8 dienu intervālā
$event = "SELECT * FROM UFC_Single_Event
WHERE eventDate >= CURDATE() AND eventDate <= DATE_ADD(CURDATE(),
INTERVAL 8 DAY)
ORDER BY id DESC";
$stmt = $conn->prepare($event);
// Pārbaude priekš veiksmīgas izpildes
if ($stmt) {
    // Izpilda vaicājumu
    $stmt->execute();
    // Iegūst rezultātus no vaicājuma un ieraksta mainīgajā $result
    $result = $stmt->get_result();
    $i = 0;
    $eventPrinted = false;
    $cardNull = null;
    // Kamēr ir rindas, izvada datus uz ekrāna
    while ($rowEv = $result->fetch_assoc()) {
        $figAway = $rowEv["fighter_away_id"];
        $figHome = $rowEv["fighter_home_id"];
        //Izvēlas datus no tabulas Fighter par attiecīgajiem cīkstoņiem (home un away)
        $fighterHome = "SELECT * FROM Fighter WHERE fig_id = $figHome";
        $fighterAway = "SELECT * FROM Fighter WHERE fig_id = $figAway";
        $figHomeSelected = select($fighterHome, $conn);
        $figAwaySelected = select($fighterAway, $conn);
        $figHomeSelected = $figHomeSelected->fetch_assoc();
        $figAwaySelected = $figAwaySelected->fetch_assoc();
        //Veic labojumus datiem, lai izvadītu uz ekrāna pareizi formatētus
        $fighterHomeFull = substr(strstr($figHomeSelected["fighter"], ','), 2) . ' .
        strstr($figHomeSelected["fighter"], ',' , true);
        $fighterHomeRank = str_replace('99', 'NR', ($figHomeSelected["rank"]));
        $fighterHomeRank = ($figHomeSelected["rank"] === '0') ? 'C' : str_replace('99', 'NR',
        $figHomeSelected["rank"]);
        $fighterAwayFull = substr(strstr($figAwaySelected["fighter"], ','), 2) . ' .
        strstr($figAwaySelected["fighter"], ',' , true);
        $fighterAwayRank = ($figAwaySelected["rank"] === '0') ? 'C' : str_replace('99', 'NR',
        $figAwaySelected["rank"]);
        $fightBout = ucfirst(str_replace('lightheavy', 'light heavyweight', str_replace('_', ' ',
        $rowEv["weightDiv"])));
```

```

$eventId = $rowEv["event_id"];
//Izvēlas datus no tabulas eventInfo par attiecīgajām cīņām
$eventId = "SELECT * FROM eventInfo WHERE competitionId = $eventId";
$eventSelected = select($eventId, $conn);
$eventSelected = $eventSelected->fetch_assoc();
$currentEvent = $eventSelected["event"];
$cardType = $rowEv["cardType"];
$bets = "SELECT * FROM UserBets WHERE SingleEventId = " . $rowEv["id"] . " AND
userId = $userId";
$currentBet = select($bets, $conn);
$currentBet = $currentBet->fetch_assoc();
// echo "<pre>";
// print_r($currentBet);
?>
<?php
//Izvada pasākuma nosaukumu, ja tas nav izvadīts jau iepriekš
if (!$eventPrinted) { ?>
    <div class="eventName flex flex-col items-center justify-center">
        <h1 class="text-4xl sm:text-5xl font-bold text-[#e4c065] drop-shadow-xl" style="text-
shadow: 0 0 5px #4E4E4E, 1px 0 0 #4E4E4E,
        -1px 0 0 #4E4E4E, 0 1px 0 #4E4E4E, 0 -1px 0 #4E4E4E,
        1px 1px #4E4E4E, -1px -1px 0 #4E4E4E, 1px -1px 0 #4E4E4E,
        -1px 1px 0 #4E4E4E;"
        ><?php echo $currentEvent; ?></h1>
    </div>
    <?php
        $eventPrinted = true;
    }
?>
<?php
//Izvada kārts nosaukumu, ja tas nav izvadīts jau iepriekš
if ($cardType != $cardNull) { ?>
    <div class="flex flex-col w-full">
        <div class="cardType flex flex-col min-h-[5vh] w-full text-center justify-center items-
center font-semibold bg-[#606060] text-[#e4c065] drop-shadow-xl">
            <h2 class="text-3xl sm:text-4xl font-bold text-[#e4c065] drop-shadow-xl justify-
center items-center"
                style="text-shadow: 0 0 5px #4E4E4E, 1px 0 0 #4E4E4E,
                -1px 0 0 #4E4E4E, 0 1px 0 #4E4E4E, 0 -1px 0 #4E4E4E,
                1px 1px #4E4E4E, -1px -1px 0 #4E4E4E, 1px -1px 0 #4E4E4E,
                -1px 1px 0 #4E4E4E;"><?php echo $cardType; ?></h2>
            </div>
            <div class="flex w-full min-h-[0.5vh] bg-[#e4c065]"></div>
        </div>
        <?php
            $cardNull = $cardType;
        }
    }
}

```

```

?>
<!-- Izvada cīņu informāciju uz ekrāna pareizi formatētu un pielāgotu lietotājam-->
<div class="main1 flex flex-row min-h-[40vh] sm:min-h-[30vh] justify-center items-center
font-semibold w-full bg-[#606060] text-[#e4c065] drop-shadow-xl"
  data-mainEv="<?= $evId ?>" id="mainEv-<?= $rowEv['id'] ?>">
  <div class="Profile hidden flex-col lg:flex">
    
  </div>
  <div class="Data flex flex-col justify-center gap-4">
    <?php
      //Ja cīņa ir čempionāta cīņa, tad izvada uz ekrāna Čempionāts
      if ($rowEv["titleBout"] == "true") {
        ?><h3 class="text-2xl font-semibold text-[#e4c065]">Championship</h3><?php
      } elseif ($rowEv["titleBout"] == "false") {
        echo "";
      } ?>
    <div class="flex flex-col sm:flex-row items-center justify-around gap-4 sm:gap-16
flex-wrap">
      <div class="flex flex-row gap-4 text-lg items-center justify-center">
        <!-- Izvada cīkstoni, kā arī viņa rangū-->
        <p><?php echo $fighterHomeRank ?></p>
        <h4><?php echo $fighterHomeFull ?></h4>
        <!-- Izvada checkbox, kas ļauj lietotājam izvēlēties cīkstoni-->
        <input id="checkBoxHome-<?= $rowEv['id'] ?>" data-event="<?= $rowEv['id']
?>" type="checkbox"
          name="fight-<?= $i ?>"
          value="<?= $figHomeSelected['id'] ?>"
          class="h-5 w-5 rounded-sm accent-[#e4c065]">
      </div>
      <div class="flex flex-row gap-4 text-lg items-center justify-center">
        <!-- Izvada checkbox, kas ļauj lietotājam izvēlēties cīkstoni-->
        <input id="checkBoxAway-<?= $rowEv['id'] ?>" type="checkbox" name="fight-
<?= $i ?>"
          value="<?= $figAwaySelected['id'] ?>"
          class="h-5 w-5 rounded-sm accent-[#e4c065]">
        <!-- Izvada cīkstoni, kā arī viņa rangū-->
        <p><?php echo $fighterAwayRank ?></p>
        <h4><?php echo $fighterAwayFull ?></h4>
      </div>
    <script>
      toggleCheckboxes('checkBoxHome-', 'checkBoxAway-', <?= $rowEv['id']?>);
    </script>
  </div>
  Kā arī es izmantoju kodu php, html un tailwind kodu priekš pogām.
  <!-- Izvada pogas, kas ļauj lietotājam izvēlēties likmes summu-->
  <div class="Buttons flex flex-row gap-4 justify-center items-center flex-wrap">

```



```

<!-- Ja likmes poga ir uzspiesta un ir izvēlēts cīkstonis, tad izvada pogu ar aktīvu
klasi un vērtību noteikto-->
<!-- Ja likmes poga ir uzspiesta pārējās pogas tiek atslēgtas un tiek pievienota ikona, lai
norādītu to-->
<!-- Ja likmes poga nav nospiesta, tad visām pogā tiek pieskirta to sākuma vērtība-->
<button value="10"
    id="bet-<?= $rowEv['id'] ?>-10"
    type="button"
    data-amount="10"
    onclick="activateButton(<?= $rowEv['id'] ?>,10)"
    class="currency-btn-<?= $rowEv['id'] ?> rounded bg-[#4E4E4E] text-[#e4c065] font-bold
w-20 h-[2rem] text-xl
    hover:bg-[#e4c065] hover:text-[#4E4E4E] transition duration-150 ease-out hover:ease-in
    hover:transition duration-300 ease-out
    <?php if (isset($currentBet['BetAmount']) && $currentBet['BetAmount'] == 10) {
        echo "active";
    } elseif (isset($currentBet['BetAmount']) && $currentBet['BetAmount'] != 10) {
        echo "disabled";
        echo "uil uil-ban text-red-500";
    }
    ?>
    "
>
<?php
if (isset($currentBet['BetAmount']) && $currentBet['BetAmount'] == 10) {
    echo "10";
} elseif (isset($currentBet['BetAmount']) && $currentBet['BetAmount'] != 10) {
    echo "";
} else {
    echo "10";
}
?>
</button>
<?php
    $i++;
}
$stmt->close();
} else {
    // Ja nav neviena cīkstona, tad izvada kļūdas paziņojumu
    echo "Sorry we ran into an error!";
}
$conn->close();
    Javascript AJAX priekš datu padošanas uz php failu, kas ievieto tos datu bāzē.

//Funkcija, kas padod iegūtos datus uz updateCurrency.php
function updCoin(newCoin, fighter, event, koef, mainEv) {
    let xhttp = new XMLHttpRequest()

```

```

xhttp.open('POST', 'controllers/updateCurrency.php', true)
xhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) {
        let response = this.responseText;
    }
};
let data = 'coin=' + newCoin + '&fighter=' + fighter + '&event=' + event + '&koef=' + koef +
'&mainEv=' + mainEv;
xhttp.send(data);
console.log('data=' + data);
}

```

Php koda daļa, lai ievietotu datus datu bāzē.

```

// Coin ir tā vērtība pogas, kuru lietotājs nospiežot, tiek ievadīta datubāzē pie BetsAmount
if (isset($_POST['coin'])) {
    session_start();
    $userID = $_SESSION['user'];
    $currency = $_POST['coin'];
    $eventID = $_POST['event'];
    $fighterID = $_POST['fighter'];
    $koef = $_POST['koef'];
    $mainEv = $_POST['mainEv'];
    // Dabū lietotāja kopējo valūtas summu
    $sqlGetCurrency = "SELECT currency FROM loginhelp WHERE id = ?";
    // Aizsargā pret SQL injekcijām
    $stmtGetCurrency = $conn->prepare($sqlGetCurrency);
    $stmtGetCurrency->bind_param("i", $userID);
    $stmtGetCurrency->execute();
    $resultGetCurrency = $stmtGetCurrency->get_result();
    $rowGetCurrency = $resultGetCurrency->fetch_assoc();
    $userTotalCurrency = $rowGetCurrency['currency'];
    // Ievieto lietotāja likmi un likmes datus datubāzē
    $sqlIns = "INSERT INTO UserBets (SingleEventId, eventId, FighterId, Koef, UserId,
BetAmount) VALUES (?, ?, ?, ?, ?, ?)";
    // Aizsargā pret SQL injekcijām
    $stmt = $conn->prepare($sqlIns);
    $stmt->bind_param("iiidii", $eventID, $mainEv, $fighterID, $koef, $userID, $currency);
    $stmt->execute();
    // Atjaunina lietotāja valūtas summu atņemot no tās likmes vērtību
    $sql = "UPDATE loginhelp SET currency = currency - ? WHERE id = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("ii", $currency, $userID);
    $stmt->execute();
    $updateCurrency = $stmt->affected_rows;
    echo json_encode($updateCurrency);
}

```

7. pielikums. Likmju rezultātiem porgrammas kods

Es izveidoju kodu, kas pēc rezultātu iegūšanas atjaunina lietotāju virtuālās naudas kopsummu. Šis notiek nepārlādējot vietni, lai lietotājs ienākot vietnē varētu redzēt naudas papildinājumu.

Es izmantoju *Javascript AJAX*, lai dati tiktu padoti tālāk uz *php* un tie atjaunotos bez lapas pārlādes.

```
//Funkcija, kas padod lietotāja atjaunināto valūtu uz updateUserMoney.php
function updCurrency() {
    let test = new XMLHttpRequest();
    test.open('POST', 'controllers/updateUserMoney.php', true);
    test.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    test.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            let response = this.responseText;
            console.log("answer from ajax: " + response);
            if (response !== "") {
                document.getElementById('currency').innerHTML = response;
            }
        }
    };
    let data = 'getResult=1';
    test.send(data);
}
updCurrency();
```

Lai atjauninātu rezultātus es izmantoju *php*. Ja ir padots *getResult*, tad izpildas noteiktais kods. Kā arī viss šis kods atrodas iekšpus *while* cikla.

```
// Ja no javascript ir padota vērtība getResult tad izpilda kodu
if (isset($_POST['getResult'])) {
    $userID = $_SESSION['user'];
    // Paņem no datubāzes visu, kas ir vienāds ar lietotāja ID
    $stmt = $conn->prepare("SELECT * FROM UserBets WHERE UserId = ?");
    // Aizsargā pret SQL injekcijām
    $stmt->bind_param("i", $userID);
    $stmt->execute();
    $result = $stmt->get_result();
    // Nosacījums pārbauda, vai vaicājumā atgriezto rindu skaits ir lielāks par nulli.
    if ($result->num_rows > 0) {
        $hasResultsThisWeek = false;
        // Iegūst no datubāzes ierakstus vienu pa vienam
        while ($row = $result->fetch_assoc()) {
            $fighterId = $row['FighterId'];
```

```

$betAmount = $row['BetAmount'];
$koef = $row['Koef'];
$mainEv = $row['eventId'];
// Paņem no datubāzes visu, kas ir vienāds ar id
$fighterNameStmt = $conn->prepare("SELECT * FROM Fighter WHERE id = ?");
// Aizsargā pret SQL injekcijām
$fighterNameStmt->bind_param("i", $fighterId);
$fighterNameStmt->execute();
$fighterNameResult = $fighterNameStmt->get_result();
$fighterNameResult = $fighterNameResult->fetch_assoc();
$fighterName = $fighterNameResult['fighter'];
$figId = $fighterNameResult['fig_id'];
$fighterNameStmt->close();
// Ievāc visus datus no ufcResults tabulas, kur eventId, singleEventId un date tiek iegūti 6
dienu intervālā
$result2 = $conn->prepare("SELECT * FROM ufcResults WHERE eventId = ? AND
singleEventId = ? AND `date` >= CURDATE() - INTERVAL 6 DAY AND `date` <=
CURDATE()");
// Aizsargā pret SQL injekcijām
$result2->bind_param("ii", $mainEv, $row['SingleEventId']);
$result2->execute();
$result2 = $result2->get_result();
// Nosacījums pārbauda, vai vaicājumā atgrieztu rindu skaits ir lielāks par nulli.
if ($result2->num_rows > 0) {
    $hasResultsThisWeek = true;
    $row2 = $result2->fetch_assoc();
    $fightWinner = $row2['fightWinner'];
    // Aprēķina likmes izmaksu. Koeficients reizināts ar likmes daudzumu un dalīts ar 20
    $calculate = intval($koef * $betAmount / 20);
    // Ja rinda paid = 0, tad izpilda kodu
    if ($row['paid'] == 0) {
        // Atjaunina lietotāja tobrīdējo naudas daudzumu ar aprēķināto likmes izmaksu
        $updateStmt = $conn->prepare("UPDATE loginhelp SET currency = currency + ?
WHERE id = ?");
        $updateStmt->bind_param("ii", $calculate, $userID);
        $updateResult = $updateStmt->execute();
        // Ja mainīgais $updateResult ir izsaukts, tad izpilda kodu
        if ($updateResult) {
            // Atjaunina paid = 1, lai vairs nevarētu izmaksāt likmi
            $updatePaidStmt = $conn->prepare("UPDATE UserBets SET paid = 1 WHERE
UserId = ?");
            $updatePaidStmt->bind_param("i", $userID);
            // Novērš sql injekcijas
            $updatePaidStmt->execute();
            // Iegūst lietotāja id no tabulas loginhelp
            $playerPaidStmt = $conn->prepare("SELECT * FROM loginhelp WHERE id =
?");

```

```

        $playerPaidStmt->bind_param("i", $userID);
        // Novērsš sql injekcijas
        $playerPaidStmt->execute();
        $playerPaidStmt = $playerPaidStmt->get_result();
        $playerPaidStmt = $playerPaidStmt->fetch_assoc();
        //Izsauc lietotāja naudas daudzumu
        echo $playerPaidStmt['currency'];

    }

}

}
$result2->close();
}
}
$result->free_result();
$stmt->close();
$conn->close();
}

```