

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Leonid Demidov 213641IABB, Vladislav Suprun 213654IABB

SQL ülesannete iseseisva harjutamise veebikeskkonna loomine

Bakalaureusetöö

Juhendaja: Erki Eessaar
PhD

Tallinn 2024

Autorideklaratsioon

Kinnitame, et oleme koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autorid: Leonid Demidov, Vladislav Suprun

20.05.2024

Annotatsioon

Lõputöö eesmärk on luua Tallinna Tehnikaülikoolis õppijatele veebipõhine keskkond SQL ülesannete iseseisvaks lahendamiseks ning selle kaudu SQL keele harjutamiseks. Keskkond toetab erineva raskusastmega ülesandeid ning erinevat tüüpi SQL lauseid (SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE ja DROP TABLE) ja pakub automaatset tagasisidet, aidates õppijatel arendada SQL oskusi PostgreSQL'i SQL keele dialekti põhjal.

Uus lahendus ületab olemasolevate veebikeskkondade piirangud, mis keskenduvad peamiselt SELECT lausetele ning pakuvad piiratud arvu ülesandeid ja vähe funktsionaalsust. Töö käigus analüüsiti olemasolevaid lahendusi, määratleti uue keskkonna nõuded ja töötati välja kasutajaliidese prototüüp, mida testiti kasutajate tagasiside alusel.

Töö tulemusena valmis PHP keeles kirjutatud funktsioneeriv veebirakendus, mis võimaldab lahendada SQL ülesandeid erinevatel andmebaasidel, saada kohest tagasisidet ja vihjeid ning jälgida edusamme. Lõputöö annab ülevaate rakenduse loomisest ning valideerimisest ja pakub soovitusi edasiseks arenduseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 78 leheküljel, 6 peatükki, 44 joonist, 5 tabelit.

Abstract

Creating a Web Environment for the Self-Practicing of SQL Tasks

The thesis aims to address the growing need for effective SQL training tools, particularly for students at Tallinn University of Technology. SQL is a crucial skill in database management, essential for the rapidly expanding digital economy. Current online platforms for practicing SQL statements often have limitations, such as focusing only on SELECT statements or offering a limited number of tasks, none of which is in Estonian.

The objective of this thesis is to develop a comprehensive web-based environment where students can independently solve SQL tasks by writing SQL statements using the PostgreSQL SQL dialect. The platform is designed to provide diverse tasks of varying difficulty levels, based on different databases, and offer automatic feedback and hints to support the learning process. The application logs attempts to solve tasks, making it possible to use the data for detailed analysis and progress tracking. The platform has been created by using PHP (8) language.

Key features of the developed application include user authentication via a learning management system, selection of language of user interface and content (currently Estonian and English are supported), a robust task search and filtering mechanism, a SQL code editor with syntax highlighting and autocompletion as well as possibility to view task history and statistics of task attempts. The system supports tasks that require writing SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE, and DROP TABLE statements.

Creating the user interface for the administrators was not a goal of the thesis. However, the system is database-based. Thus, administrators can easily add new databases and tasks to meet the needs of users.

The development process followed an inductive approach, incorporating feedback from students and the database course instructor, who also served as the thesis supervisor. The

system was iteratively evaluated and refined based on user feedback, resulting in a reliable and user-friendly tool.

In conclusion, the web application provides a valuable resource for students to improve their SQL skills through independent practice. It should significantly enhance the learning experience, offering immediate feedback and a wide range of practice opportunities. This tool not only supports the academic curriculum but also prepares students for professional challenges in data management and analysis.

The thesis is in Estonian and contains 78 pages of text, 6 chapters, 44 figures, 5 tables.

Lühendite ja mõistete sõnastik

AJAX	<i>Asynchronous JavaScript and XML</i> . Meetod veebilehtede interaktiivsemaks muutmiseks, võimaldades asünkroonset andmevahetust veebilehe ja serveri vahel ilma kogu lehe uuesti laadimiseta.
API	<i>Application Programming Interface</i> . Liides, mis võimaldab erinevatel tarkvararakendustel omavahel suhelda ja andmeid vahetada.
CSRF	<i>Cross-Site Request Forgery</i> . Veebirünnaku tüüp, kus pahatahtlik kasutaja petab ohvri veebilehitsejat tegema soovimatuid toiminguid teise veebisaidi nimel, millele kasutaja on autenditud.
HTML5	<i>HTML5</i> on <i>HTML (HyperText Markup Language)</i> standardi viies ja hetkel kehtiv versioon, mida kasutatakse veebilehtede ja veebirakenduste struktuuri ja sisu loomiseks. HTML5 tutvustab mitmeid uusi elemente, atribuute (header, footer, article, section, nav jne) ja API-sid (Drag and Drop, WebSockets jne), mis võimaldavad arendajatel luua rikkalikumaid ja interaktiivsemaid veebikogemusi.
HTTP	<i>HyperText Transfer Protocol</i> . See on veebiserverite ja veebilehitsejate vaheline protokoll, mis võimaldab veebilehtede edastamist ja suhtlust klientidega internetis
HTTPS	<i>HyperText Transfer Protocol Secure</i> . Turvaline versioon HTTP-st, mis kasutab andmete krüpteerimiseks SSL/TLS protokoll, tagades seega turvalise andmevahetuse veebiserveri ja -lehitseja vahel.
Lõpp-punkt	Konkreetne URL või aadress API-s, millele saab teha päringuid ja mille kaudu saab andmeid saata või vastu võtta.
Modaalaken	<i>Modaalaken</i> on veebilehtede ja tarkvararakenduste kasutajaliidese element, mis ilmub esiplaanile ja nõuab kasutajalt mingi tegevuse sooritamist või otsuse tegemist enne, kui ta saab naasta põhiakna või -lehe juurde.
MVC	<i>Model-View-Controller</i> . Tarkvara arhitektuuri muster, mis eraldab andmete esituse, kasutajaliidese ja äriloogika erinevatesse komponentidesse, et parandada rakenduse struktuuri ja hooldatavust.

REST	<i>Representational State Transfer</i> . Arhitektuuristiil API-de kujundamiseks, mis kasutab HTTP protokollid ning määratleb selged reeglid ressursside loomiseks, lugemiseks, uuendamiseks ja kustutamiseks.
Sait	<i>Sait</i> on veebileht või veebilehtede kogum, mis on majutatud ühel veebiaadressil ja on kättesaadav interneti kaudu. Seda kasutatakse teabe jagamiseks, teenuste pakkumiseks või kasutajatega suhtlemiseks.
SVG	<i>Scalable Vector Graphics</i> . SVG on XML-põhine vektorgraafika formaat, mida kasutatakse graafilise sisu loomiseks ja kuvamiseks veebilehtedel.
SQL	<i>Structured Query Language</i> . Standardiseeritud andmebaasikeel, mida kasutatakse relatsiooniliste andmebaaside haldamiseks ja nendes olevate andmete töötlemiseks.
URL	<i>Uniform Resource Locator</i> . See on veebiaadress, mis identifitseerib konkreetset ressursi internetis. See sisaldab protokollid (nt HTTP või HTTPS), domeeninime või IP-aadressi ning ressursi täpset asukohta serveris

Sisukord

1 Sissejuhatus	14
1.1 Lahendatav probleem	15
1.2 Töö edasine struktuur	15
2 Metoodika.....	17
2.1 Tööprotsessi kirjeldus.....	19
2.2 Tööriistade kirjeldus	20
3 Olemasolevad SQLi automaatkontrolli lahendused	23
3.1 Olemasolev tarkvara	24
3.1.1 HackerRank	24
3.1.2 LeetCode.....	25
3.1.3 PostgreSQL Exercises	26
3.1.4 SQLPad.....	27
3.1.5 Interaktiivne veebiõpik Tartu Ülikooli õppeainele " Sissejuhatus andmebaasidesse"	29
3.2 Teaduskirjandus.....	30
4 Töö tulemused	33
4.1 Tarkvara funktsionaalsus.....	33
4.1.1 Autoriseerimine Mauruse süsteemi kaudu	33
4.1.2 Keele valik.....	34
4.1.3 Ülesande otsimine	34
4.1.4 Ülesande lahendamine.....	35
4.1.5 Ülesannete lahendamise ajalugu.....	38
4.1.6 Ülesande lahendamise ajalugu	38
4.1.7 Statistika	39
4.2 Mittefunktsionaalsed omadused	40
4.2.1 Esitus	40
4.2.2 Turvalisus	41
4.2.3 Töökindlus.....	42
4.2.4 Hooldatavus.....	43

4.3 Kasutajaliides.....	44
4.3.1 Värvipalett	44
4.3.2 Interaktiivsus	44
4.3.3 Informatiivsus.....	46
4.3.4 SQL kood.....	46
4.3.5 Kasutajaliidese kohandamine lõppkasutaja poolt.....	47
4.3.6 Ligipääsetavus	47
4.3.7 Kasutajaliidese kohandamine administraatori poolt.....	49
4.4 Arendusprotsess ja tulemused	50
4.4.1 Valitud tehnoloogiate kooskõlastamine	50
4.4.2 Graafilise prototüübi loomine.....	51
4.4.3 Tagasiside kogumine prototüübi kohta	52
4.4.4 Eesrakenduse arendus.....	52
4.4.5 Välised teegid ja komponendid	52
4.4.6 SQL koodi kontrollimine.....	53
4.4.7 Tagarakenduse arendus	57
4.4.8 Ees- ja taga rakenduse ühendamine.....	57
4.4.9 Testimine	57
4.4.10 Väljalase ja tagasiside kasutajatelt	58
5 Analüüs ja järeldused.....	60
5.1 Võrdlus olemasoleva tarkvaraga	60
5.1.1 Kasutajaliidese eelised võrreldes olemasoleva tarkvaraga.....	60
5.1.2 Funktsionaalsuse eelised võrreldes olemasoleva tarkvaraga.....	61
5.1.3 Funktsionaalsuse puudused võrreldes olemasoleva tarkvaraga	62
5.2 Kasutajate poolne kasutajaliidese prototüübi testimine.....	63
5.2.1 Küsitluse korraldus ja küsimustik	63
5.2.2 Tulemused	64
5.2.3 Tulemuste analüüs ja nende alusel tehtud muudatused.....	65
5.3 Kasutajate poolne rakenduse testimine.....	66
5.3.1 Küsitluse korraldus ja küsimustik	66
5.3.2 Tulemused	67
5.3.3 Tulemuste analüüs	68
5.3.4 Tulemuste analüüsi alusel tehtud muudatused	70
5.4 Töö tegemise refleksioon	71

5.4.1 Asjad, mis läksid hästi.....	71
5.4.2 Asjad, mis läksid halvasti.....	72
5.4.3 Asjad, mida teha tööd korrates teistmoodi.....	72
5.4.4 Tehtud töö kasulikkus.....	72
5.5 Edasine arendus.....	74
6 Kokkuvõte.....	76
Kasutatud kirjandus.....	77
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	79
Lisa 2 – Lõpp-punktide nimekiri.....	80
Lisa 3 – Võimalikud vead kasutaja koodi kontrollimisel.....	82
Lisa 4 – Ülesannete registrite andmebaasi disaini skeemid.....	84
Lisa 5 – Kasutajaliidese illustatsioonid.....	86
Lisa 6 – Prototüübi küsimustik.....	90
Lisa 7 – Figma prototüüp.....	94
Lisa 8 – Prototüübi uuringu tulemustel põhinevad tulpdigrammid.....	97
Lisa 9 – Veebirakenduse uuringu tulemustel põhinevad tulpdigrammid.....	100
Lisa 10 – Olemasolevate SQLi harjutamiste rakenduste kasutajaliides.....	103
Lisa 11 – Olemasolevate vahendite võrdlustabel.....	105
Lisa 12 – Leonid Demidovi panuse kirjeldus ja eneseanalüüs.....	107
Lisa 13 – Vladislav Suprun panuse kirjeldus ja eneseanalüüs.....	109

Jooniste loetelu

Joonis 1. Config.php fail.	50
Joonis 2. Tabelite B ja A loomine vastavalt kasutaja otsusele ja probleemi võimalikule lahendusele.	56
Joonis 3. Hinnangud lehtede kohta.	65
Joonis 4. Vastajate SQL-oskuse enesehinnang.	67
Joonis 5. Sektordiagramm tulevase kasutamise kohta.	68
Joonis 6. Sektrordiagramm platvormi soovitamise kohta.	68
Joonis 7. Ülesannete register andmebaasi disaini skeem Enterprise Architect modelleerimisvahendis.	84
Joonis 8. Ülesannete register andmebaasi disaini skeem Enterprise Architect modelleerimisvahendis.	85
Joonis 9. Pealeht.	86
Joonis 10. Leht ülesannete loendiga.	86
Joonis 11. Ülesande täitmise leht enne viimaste täienduste tegemist.	87
Joonis 12. Ülesande edenemise leht avatud vihjete ja võimalike lahendustega.	87
Joonis 13. Ülesande täitmise leht, kui katse ebaõnnestub.	88
Joonis 14. Ülesande täitmise ajaloo leht.	88
Joonis 15. Kõigi ülesannete ajaloo leht.	89
Joonis 16. Statistika leht.	89
Joonis 17. Prototüübi küsimustiku teine leht.	90
Joonis 18. Prototüübi küsimustiku kolmas leht.	91
Joonis 19. Prototüübi küsimustiku neljas leht.	92
Joonis 20. Prototüübi küsimustiku viimane leht.	93
Joonis 21. Pealehe prototüüp Figma.	94
Joonis 22. Ülesannete ja kogumi loendi lehe prototüüp Figma.	94
Joonis 23. Ülesande lehe prototüüp Figma.	95
Joonis 24. Ajaloo lehe prototüüp Figma.	95
Joonis 25. Ülesande ajaloo lehe prototüüp Figma.	96
Joonis 26. Statistika lehe prototüüp Figma.	96

Joonis 27. Hinnangud pealehe kohta.	97
Joonis 28. Hinnangud ülesannete loendi kohta.	97
Joonis 29. Hinnangud ülesanne lahendamise lehe kohta.....	98
Joonis 30. Hinnangud ajaloo lehe kohta.	98
Joonis 31. Hinnangud ülesande ajaloo lehe kohta.	99
Joonis 32. Hinnangud statistika lehe kohta.	99
Joonis 33. Hinnangud pealehe kohta.	100
Joonis 34. Hinnangud ülesannete loendi lehe kohta.....	100
Joonis 35. Hinnangud ülesande lahendamise lehe kohta.....	101
Joonis 36. Hinnangud ülesannete ajaloo lehe kohta.....	101
Joonis 37. Hinnangud ülesande ajaloo lehe kohta.	102
Joonis 38. Hinnangud statistika lehe kohta.	102
Joonis 39. HackerRank ülesannete loendi leht.	103
Joonis 40. LeetCode platvormi ülesande lahenduse leht, kus tabelid on tekstivormis.	103
Joonis 41. PostgreSQL Exercises ülesande täitmise leht.	104
Joonis 42. SqlPad ülesannete loendi leht.....	104
Joonis 43. Leonid Demidovi tööks kulunud aja jaotus.....	107
Joonis 44. Vladislav Supruni tööks kulunud aja jaotus.	109

Tabelite loetelu

Tabel 1. HTML Checkeriga lehtede kontrollimise tulemus.....	48
Tabel 2. Lehtede kirjeldus.	51
Tabel 3. Lõpp-punktide nimekiri.....	80
Tabel 4. Võimalikud vead kasutaja koodi kontrollimisel.....	82
Tabel 5. Olemasolevate vahendite võrdlustabel.	105

1 Sissejuhatus

Andmebaasikeel SQL (*Structured Query Language*) on oluline tööriist andmebaaside halduses, mille kasutamise oskus on üha kasvavas digitaalses majanduses väga vajalik. Tänapäeva maailmas, kus andmete hulk kasvab eksponentsiaalselt ja informatsioonist on saanud peamine ressurss, on SQL-i tähtsus suurem kui kunagi varem. SQL ei ole lihtsalt programmeerimiskeel. See keel on võti, mis avab andmete kasutamise võimaluse, muutes toorandmed väärtuslikuks informatsiooniks, mis on otsuste tegemisel ja äristrateegiate kujundamisel kriitilise tähtsusega.

Digitaalajastul, kus ettevõtted ja organisatsioonid sõltuvad suuresti andmete kiirest analüüsist ja töötlemisest, on oskus kirjutada SQL-is keerukaid päringuid ja teha andmemuudatusi, muutunud hädavajalikuks. Olgu tegemist kliendi käitumise mustrite analüüsimise, finantstulemuste prognoosimise või operatiivsete riskide maandamisega, SQL võimaldab organisatsioonidel hallata suuri andmekogumeid efektiivselt ja täpselt. Vaatamata sellele, et keelt hakati välja töötama juba 1970-ndatel ja selle keele rahvusvahelise standardi esimene versioon ilmus 1986. aastal, on SQL endiselt arenev keel, kuhu iga uue keele standardi väljalaskega lisandub uusi võimalusi [1].

Lisaks on SQL-i tähtsust rõhutatud mitmetes eluvaldkondades, alates tervishoiust ja finantssektorist kuni tehnoloogia ja valitsusasutusteni, sest see on suurandmete (andmeid on palju, neid tuleb nagu kosest juurde ja need on erineva formaadiga) ajastul asendamatu tööriist. Tänu võimekusele toetada kriitilisi ärifunktsioone ja operatsioone on SQL osutunud keskseks lüliks edukate ettevõtete tehnoloogilises arsenalis. IEEE 2023. aasta programmeerimiskeelte populaarsuse indeksi kohaselt olid SQL-i oskused tööturul väga nõutud [2], peegeldades selle keele olulisust ja universaalsust kiiresti muutuval majandusmaastikul. Seega, SQL-i mõistmine ja valdamine on mitte ainult soovitatav, vaid tihti ka karjääri edendamiseks kriitilise tähtsusega oskus, mis avab uksi arvukates sektorites üle kogu maailma.

1.1 Lahendatav probleem

Tegevustes vilumuse saavutamiseks, oskuste kinnistamiseks ning tehtava sisuliseks mõistmiseks on oluline järjepidev harjutamine. Harjutamisel on oluline saada kohest tagasisidet, kuid kui tehtud töö kontrollijad on inimesed, siis see seab kontrollitavate ülesannete arvule ja tagasiside andmise kiirusele ette piirid. Seega oleks vajalik lahendus, kus kontrollimine toimuks automaatselt, tarkvara poolt. SQL-i iseseisev harjutamine on oluline, kuid olemasolevad veebikeskkonnad ei vasta täielikult õppijate vajadustele. Olemasolevad veebikeskkonnad ei vasta nende vajadustele, kuna enamik neist keskendub ainult SELECT lausetele ja pakuvad piiratud arvu ingliskeelseid ülesandeid. Lisaks on platvormide võimalused uute ülesannete lisamiseks piiratud või need on tasulised ja suletud süsteemid.

Seega on eesmärgiks luua veebipõhine keskkond, mis võimaldab Tallinna Tehnikaülikooli üliõpilastel iseseisvalt harjutada erinevat tüüpi SQL lausete kirjutamist PostgreSQL dialekti e mägimurraku põhjal, mis toetab paljusid uusimaid SQLi võimalusi. Vahend peaks võimaldama pakkuda mitmekülgeid, erineva raskusastmega ja erinevatel andmebaasidel põhinevaid ülesandeid. Oluline on tagada automaatne tagasiside andmine ja lahendamise vihjete pakkumine, et toetada õppimist ja oskuste arendamist. Samuti peaks platvorm logima ülesannete lahenduskatseid, võimaldades edaspidi nende analüüsi ja õppijate edusammude jälgimist. Oodatavaks tulemuseks on funktsioneeriv veebirakendus, mis aitab parandada SQLi oskuseid iseseisva harjutamise kaudu. Töö keskendub SQL lausete harjutajale mõeldud kasutajaliidese loomisele, samal ajal kui ülejäänud süsteemi haldamisega tegeleb administraator. Administraatorile mõeldud kasutajaliidese loomine ei ole käesoleva töö ülesanne Rakendus on plaanitud arendada PHP programmeerimiskeeles kasutades PostgreSQL andmebaasisüsteemi. Käesoleva töö juhendaja, kes õpetab Tallinna Tehnikaülikoolis andmebaaside kursuseid, on ühtlasi ka antud tarkvara tellija ning administraatori rollis.

1.2 Töö edasine struktuur

Töö on jaotatud mitmeks peamiseks etapiks, alustades olemasolevate süsteemide analüüsist ja puuduste tuvastamisest. Sellele järgneb uue keskkonna funktsionaalsete ning mittefunktsionaalsete nõuete määratlemine ning kasutajaliidese prototüübi arendamine koos esialgse tagasiside kogumisega. Peale selle keskendutakse arendamisele ja

testimisele, võimaldades süsteemi pidevat täiustamist vastavalt kasutajate tagasisidele ja uutele esilekerkinud nõuetele. Järgneb rakenduse lõplik testimine enne selle võrdlemist olemasolevate tarkvaradega. Analüüsi ja järelduste osas hinnatakse kasutajate tagasisidet, töö tegemise protsessi ning antakse soovitusi edasiseks arenduseks. Töö kokkuvõttes esitatakse peamised järeldused ja tulemused.

2 Metoodika

Lõputöö tegemiseks kasutatakse induktsiooni meetodit [3]. See valik on tingitud vajadusest luua uut tarkvara, täpsemalt töötavat programmi Tallinna Tehnikaülikooli üliõpilastele. Induktiivne lähenemine võimaldab uurida konkreetseid näiteid ja juhtumeid (antud juhul erinevaid SQLi harjutamise keskkondi) üldiste seaduspärasuste ja mustrite tuvastamiseks, mis on võtmetähtsusega toote nullist loomise juures. Selline meetod võimaldab sügavamalt mõista sihtgrupi vajadusi ja eelistusi, mis omakorda võimaldab luua täpsema ja efektiivsema lahenduse. Lisaks võimaldab induktsiooni kasutamine analüüsida ja võrrelda olemasolevaid lahendusi ja platvorme, et tuvastada nende eelised ja puudused, mis on oluline tootearenduse jaoks, mis rahuldaks maksimaalselt lõppkasutajate vajadusi.

Töös loodi uus tehis (SQLi harjutamise keskkond) arvestades nii valdkonna parimate praktikate kui kasutajate nõuetega ja seejärel valideeriti seda reaalses kasutusolukorras. Tehtud töö valideerimiseks pandi see edukalt tööle ülikooli serveris ja anti juurdepääs kõigile ülikooli õppeaine “Andmebaasid I” üliõpilastele, et saada nendelt tagasisidet. Kuna tehise nii tehise kasutajaliidese prototüübi kui esimese versiooni loomise järel koguti selle kohta hulgaliselt tagasisidet ning selle alusel tehti tehisesse veel parandusi, siis kokkuvõttes on tegemist disaini tegevusuuringu (*action design research*) [4] stiilis tööga, kus tehise hindamine toimub tehise arendamisega samaaegselt ning selle tulemused mõjutavat tehise arendust.

Arendustöös kasutatakse pigem koskstiili kui iteratiivset stiili. Koskstiili iseloomustab arendusetappide järjestikuline läbimine: alustades nõuete määratlemisest jätkates tehnilise kavandamisega ning seejärel liikudes realiseerimise ja siirde etappideni. Iga järgnev etapp algab alles pärast eelneva lõpetamist. Käesolevas töös pöörduiti vajadusel tagasi ka eelmiste etappide juurde. [5]

Koskstiili kasutamise oluliseks eeliseks on selle struktureeritus ja etappide selge määratlemine, mis on eriti väärtuslik projektide puhul, mis vajavad selget planeerimist ja kontrolli. Käesoleva töö puhul, kus arendatakse tarkvara Tallinna Tehnikaülikooli

üliõpilastele, tagab selline lähenemine võimaluse nõuete selgeks määratlemiseks ja funktsionaalsuse järjepidevaks rakendamiseks, mis on oluline kvaliteetse ja stabiilse toote loomisel.

Lisaks aitab koskstiil vältida olukordi, kus hilisemad arendusetapid muutuvad keerulisteks ja ajakulukateks. Seda aitab vältida nõuete põhjalik analüüs ja määratlemine projekti algfaasis. See võimaldab arendajatel minimeerida riske ja tagada lõpptoote hea vastavus kasutajate ootustele ja vajadustele. Rohkem üksikasju tööprotsessi etappidest on kirjeldatud jaotises 2.1. Tööprotsessi kirjeldus. Käesolevas töös oli selline lähenemine võimalik kuna juhendajal, kes ühtlasi oli ka tellija rollis, oli pikaajaline kogemus SQLi õpetamisega ja üsna selge nägemus sellest, mida see süsteem peaks võimaldama teha.

Lõputöö käigus tehti valikuid toetudes subjektiivsele analüüsile ja ekspertarvamusele. See lähenemine võimaldas tuvastada kriitilisemad ülesanded, mis vajavad kiiret lahendamist. Autorid analüüsisid pidevalt projekti edenemist ja kohandasid algset plaani ja prototüüpi, püüdes leida tasakaalu arendamise kiiruse ja veebisaidi tellija ning süsteemi kasutavate üliõpilaste kõrge rahulolu tagamise vahel. Selline lähenemine võimaldas neil kiiresti reageerida muutustele ja tagada projekti edukas elluviimine lühikese aja jooksul.

Pärast arenduse lõpetamist ja kasutajatel tagasiside saamist viidi läbi põhjalik tehtud töö ja saavutatud tulemuste analüüs. See protsess hõlmas täidetud ülesannete hindamist, eesmärkide saavutamise analüüsimist ning projekti tugevuste ja nõrkuste väljaselgitamist. Samuti pöörati tähelepanu kasutajate tagasisidele, et mõista nende vajadusi ja eelistusi, mis võimaldas autoritel teha objektiivseid järeldusi loodud süsteemi kvaliteedi ja tõhususe kohta. Saadud tulemusi kasutati tarkvaras täiendavate muudatuste tegemiseks, projekti lõpparuande koostamiseks (mille rollis on käesolev lõputöö) ja toote edasise täiustamise ettepanekute tegemiseks.

Lõputöö tegemise käigus järgiti rangelt eetilise teadustöö põhimõtteid. Arvestades, et juurdepääs loodud süsteemi toimub teise süsteemi (Maurus) kasutajatunnuste kaudu, pöörati erilist tähelepanu andmete turvalisusele ja konfidentsiaalsusele. Eetilise uurimistöö tagamiseks järgiti järgmisi põhimõtteid: konfidentsiaalsuse säilitamine, süsteemis hoitavate kasutajate andmete minimeerimine ja nende turvalisuse tagamine. Küsitluste tulemused ei olnud anonüümsed, kuid isikustatud vastused jäid ainult töö

autorite ja juhendaja teada. Käesolevas töös avaldatakse need peale andmetöötlust üldistatud kujul.

2.1 Tööprotsessi kirjeldus

Siin jaotisest antakse lühiülevaade protsessi sammudest. Lisateavet erinevate sammude ja nende tulemuste kohta esitatakse töö järgnevates osades.

Töö autorid said juhendajatelt sisendina süsteemi metaandmete andmebaasi, kuhu olid juba lisatud klassifikaatorite väärtused, testülesanded ja mõned SQL koodi kontrollimiseks kasutatavad funktsioonid. Lisaks said autorid ühe ülesannete lahendamiseks mõeldud andmebaasi. Samuti said töö autorid sisendina ülesannete kontrollimise loogika kirjelduse. Autorite ülesandeks oli realiseerida toimiv, mugav ja igapäevaseks õppetöökõks mõeldud kasutajaliides SQL lausete harjutajatele (st mitte prototüüp, vaid töötav tarkvara, mida saaks ülikooli õppetöös igapäevaselt kasutada).

Projekti alguses tutvuti sarnaste SQL-i iseõppimist võimaldavate süsteemide ja teemakohase teaduskirjandusega. Autorid pidasid nõu ka juhendajaga tulevase tarkvara funktsionaalsete ja mittefunktsionaalsete nõuete osas. Pärast esialgsete nõuete kokkuleppimist alustati kasutajaliidese prototüübi väljatöötamist, mida paluti hinnata "Andmebaas I" õppeaine õppijatel. Pärast tagasiside saamist tehti enne süsteemi realiseerimise alustamist prototüübis muudatusi.

Kasutajaliidese prototüübi jaoks, mis realiseeri Figmas, töötati välja mitu prototüüplehte, sealhulgas avaleht, ülesannete loendi leht, ülesannete täitmise leht, ajaloo leht, statistika leht ja programmi ülevaate leht. Samuti hoolitseti nende kolmandatele osapooltele näitamise mugavuse eest, mistõttu loodi iga lehe jaoks Figmas eraldi aken, mis võimaldas kasutajatel küljendusega mugavalt tutvuda.

Prototüübi katsetajatele tehti veebipõhine küsitlus, et saada infot nende rahulolu kohta tulevase tarkvaraga. Saadud tagasiside analüüsimise tulemusena selgitati välja parendusvajadused ning tehti vajalikud kohandused.

Arenduse võib jagada kaheks osaks: eesrakenduse ja tagarakenduse loomine. Eesrakenduse arendamiseks kasutati HTML-i, CSS-i ja JavaScripti, põhiliseks stiiliraamistikuks oli Bootstrap ning JavaScripti puhul jQuery. Tagarakenduse loomisel

realiseeriti kõik marsruudid, kasutajaseansside haldamise ja REST API mudelid ilma väliseid PHP teke või raamistikke kasutamata. Kui mõlema osa loomine olid lõpetatud, siis integreeriti kõik tagarakenduse lõpp-punktid eesrakendusse, et töödelda serveri erinevat tüüpi vastuseid, lisades kasutajakogemuse parandamiseks interaktiivseid elemente.

Tarkvara testimiseks kasutati erinevaid lähenemisviise, sealhulgas funktsionaalset testimist privaatses serveris. Testimise käigus analüüsiti rakenduse iga lehte, kontrollides iga interaktiivset elementi (modaalaknad, rippmenüü plokid, andmetabelid, lehekülgede jagamine, kerimisplokid jne), parandati tuvastatud probleemid ja pöörduti vajadusel tagasi arendusfaasi.

Seejärel lanti programm üle juhendajale, kes pani programm üles ülikooli serverisse ja konfigureeris süsteemi, määrates ära vajalikud andmed andmebaasiga ühenduse loomiseks ja saidi konfigureerimiseks. Sellele järgnevalt testis rakendust töö juhendaja, kelle tagasiside alusel tehti tarkvarasse veel mitmeid parandusi. Seejärel koguti “Andmebaasid I“ üliõpilastelt veebipõhise küsimustiku abil tagasisidet peale seda kui nad kasutasid loodud süsteemi erinevate SQL ülesannete lahendamiseks. Saadud tagasiside alusel tehti tarkvarasse veel täiendusi.

2.2 Tööriistade kirjeldus

Süsteemi loomise keeleks valiti **PHP (ver 8)**, kuna autoritel oli selle keelega juba põhiline praktiline kogemus, mis peaks võimaldama luua tarkvara piiratud ajaga ning selle keele kasutus oli ka juhendaja eelistus.

PostgreSQL (ver 16) valiti andmebaasisüsteemiks juhendaja soovil, sest tegemist on populaarse, tasuta ja SQL lausete kirjutajale palju võimalusi pakkuva andmebaasisüsteemiga, mida kasutatakse ka ülikooli andmebaaside kursustes.

Microsoft Forms on Microsofti pakutav veebiteenus, mis võimaldab luua küsitlusi, et koguda inimestelt tagasisidet. See on tuntud oma kasutajasõbralikkuse ja paindlikkuse poolest nii küsitluste loomisel kui ka andmete analüüsimisel. See platvorm võimaldab luua kohandatud küsimustikke, mis on kergesti ligipääsetavad ja vastatavad mitmesugustel seadmetel, aidates seeläbi kaasa suurema vastuste arvu kogumisele. Meeskond kasutas Microsoft Forms'i vahendina, et koguda tagasisidet prototüübi kohta

arenduse varajases faasis, samuti valminud rakenduse kohta, et saada väärtuslikke kommentaare ja ettepanekuid edasisteks arendusteks.

PgAdmin on graafiline kasutajaliides PostgreSQL'i andmebaaside haldamiseks. Töös kasutati PgAdmin'i andmebaaside kontrollimiseks, projekti algandmebaaside loomiseks ja metaandmete registreerimiseks. See vahend võimaldab andmebaasi tõhusalt hallata, tagada selle stabiilse töö ning tuvastada vead (sh koodivead) juba arenduse varajases etapis.

Docker on platvorm rakenduste loomiseks, kasutuselevõtuks ja käivitamiseks konteinerites. Meeskond kasutas Dockerit PostgreSQL'i ja PgAdmini käivitamiseks, et saavutada endale mugav andmebaasi kasutuskogemus ja tagada arenduseks kasutatud arvutites ühesuguse keskkonna olemasolu. See võimaldas hõlpsasti liikuda erinevate arenduskeskkondade vahel, ilma et oleks olnud vaja keskkonda ümber seadistada.

Openserver (Nginx_1.23, PHP 8.0) oli valik kohalikuks veebiserveriks, mida kasutati PHP-koodi käivitamiseks töö autorite kohalikel hostidel. Kuigi alternatiiviks oleks võinud olla Dockeri kasutamine koos konteinerite ja failide sünkroniseerimiseks mõeldud köidete (*volumes*) konfigureerimisega, otsustati siiski Openserveri kasuks. Selle otsuse taga oli soov näha koheselt kohalikus serveris tehtud muudatuste tulemusi. Dockeri kasutamine arendusprotsessis oleks nõudnud konteineri loomist koos PHP-ga ja köidete seadistamist failide sünkroniseerimiseks kohaliku failisüsteemi ja konteineri vahel. Kuid seadmete piiratud ressursside tõttu oleks see sünkroonimisprotsess võinud võtta märkimisväärselt palju aega, mis oleks tööd aeglustanud. Selle asemel andis Openserver muudatuste kohta kohese ülevaate, mis võimaldas keskenduda tõhusamale arendusele, ilma tarbetute viivitusteta.

Figma on veebipõhine tööriist kasutajaliidese disaini ja prototüüpide loomiseks. Töö meeskond kasutas Figmat, et luua tarkvara kasutajaliidese prototüüp ja seda mugavalt esitleda, et saada tulevastelt kasutajatelt tagasisidet. See vahend võimaldas visualiseerida tarkvara kontseptsiooni ja analüüsida tarkvara väljanägemise seisukohast.

Visual Studio Code on avatud lähtekoodiga arenduskeskkond, mis pakub laia valikut arendusvahendeid. VS Code valiti mitmel põhjusel. Esiteks, selle vahendi kiirus ja mitmekülgsus võimaldasid projekti kallal tõhusalt töötada. Samuti on VS Code'il palju laiendusi, mis tegid töö palju lihtsamaks. Siin on kõige kasulikumad, mida arenduse

käigus kasutati: “PHP File Link”, “PHP Intelephense”, “Auto Rename Tag” ja “Bootstrap Class Autocomplete”. Valmis nipid kiirendasid oluliselt koodi kirjutamise protsessi ning integreeritud tööriistade näpunäited aitasid vältida vigu ja suurendada tootlikkust. Lähtekoodis navigeerimine oli üllatavalt lihtne tänu võimalusele kiiresti olemasolevate klasside või failide juurde hüpata. Failide automaatse vormindamise funktsioon võimaldas säilitada kogu tarkvaras ühtset koodistiili, mis muutis selle kõigile meeskonnaliikmetele loetavamaks ja arusaadavamaks.

Git on versioonikontrollisüsteem. **GitLab** on veebipõhine platvorm Giti hoidlate haldamiseks. GitLabi kasutati paralleelseks arendamiseks kahel seadmel ja ülesannete harude lihtsaks ühendamiseks. Töös kasutati Giti aktiivselt koodimuudatuste kinnitamiseks, tagades projekti turvalisuse ja stabiilsuse.

WinRAR on programm failide arhiveerimiseks ja pakkimiseks. Töös kasutati WinRARi, et mugavalt ja usaldusväärset transportida faile ülikooli serverisse, tagades ülekande ajal nende terviklikkuse ja turvalisuse.

Postman on vahend veebiteenuste rakendusliideste testimiseks. Töös kasutati Postmani kõigi projekti lõpp-punktide automaatsete kirjutamiseks ja ülesannete igat tüüpi päringute testimiseks, tagamaks loodud rakenduse kvaliteedi ja töökindluse.

Enterprise Architect on tarkvara- ja infosüsteemide projekteerimise tarkvara, mida kasutatakse arenduse erinevate aspektide, näiteks äriprotsesside, andmebaasi struktuuri, rakendusarhitektuuri ja muu sellise loomiseks ning visualiseerimiseks. Töös kasutati Enterprise Architecti, et vaadata etteantud andmebaasi struktuuri .

Windows 11 on operatsioonisüsteem, millel käivitati kõik eelnimetatud utiliidid (docker, vs code, openserver, figma jne), pakkudes arendusmeeskonnale mugavat ja tõhusat töökeskkonda.

3 Olemasolevad SQLi automaatkontrolli lahendused

Selleks, et objektiivselt hinnata sarnaste süsteemide eeliseid ja puudusi, on iga süsteemi analüüsi käigus vastatud järgmistele küsimustele.

1. Kas leht on mõeldud erinevate väljapakutud ülesannete lahendamiseks või on tegemist SQLi õppematerjalide keskkonnaga, kus ülesannete lahendamine on toetav funktsioon?
2. Kas tarkvara pakutakse veebiteenusena või saaks kasutaja selle oma serverisse üles panna?
3. Milline on visuaalne üldmulje?
4. Milline on veebilehe navigeerimine (kas vajalikke jaotisi ja funktsioone on lihtne leida)?
5. Kas kasutajaliidese isikupärastamine on võimalik (teema või elementide paigutuse kohandamine)?
6. Kas on olemas intuiitiivsed juhised või nõuanded uutele kasutajatele?
7. Milline on lehtede laadimise ja toimingute tegemise kiirus saidil?
8. Kas on olemas vahendid, millega saab jälgida kasutajate edusamme SQL-i õppimisel?
9. Milline on saidi interaktiivsuse ja tagasisidevõimaluste tase (näiteks, kas on integreeritud tugifoorumid või vestlus administraatoriga)?
10. Kas on olemas võimalus vaadata vihjeid ülesannete lahendamiseks?
11. Kas on võimalus vaadata võimalikke ülesannete lahendusi?
12. Kui paljude erinevate tingimuste alusel saab ülesandeid otsida?
13. Mis tüüpi lausete (SELECT, INSERT, DELETE, MERGE jne) koostamine on toetatud?

14. Kui detailne on tagasiside vigade kohta?
15. Kas on olemas mitmekeelne kasutajaliides?
16. Millistel andmebaasisüsteemidel see tarkvara põhineb?
17. Kas SQL lausete harjutamise eest küsitakse raha?
18. Kui palju on lahendamiseks mõeldud ülesandeid?
19. Kas õppejõul oleks võimalik lisada uusi ülesandeid?

3.1 Olemasolev tarkvara

Selles jaotises analüüsitakse mõningaid olemasolevaid SQLi harjutamise keskkondi eelnimetatud küsimuste alusel.

3.1.1 HackerRank

HackerRank on veebiteenus, mis võimaldab harjutada tehniliste ülesannete, sealhulgas SQL-iga seotud ülesannete täitmist [6]. Üldiselt on saidi välimus meeldiv: sait on kujundatud minimalistlikus stiilis, värvipalett on hästi valitud ja puuduvad tarbetud elemendid (vt Joonis 39, Lisa 10). Mõned tekstipiirkonnad on aga liiga väikesed, mistõttu on neid raske lugeda. Lehtede laadimine toimub kiiresti, kuid SQL-koodi kontrollimine on isegi lihtsate päringute puhul üsna aeglane. See järeldus põhines väikesel testil, mille käigus käivitati algtaseme ülesande lahendusena järgmine lause, mis on sisuliselt õige: *“SELECT * FROM CITY WHERE COUNTRYCODE='USA' AND POPULATION > 100000”*. Samuti puudub saidil mitmekeelsuse tugi.

Saidil navigeerimine on mugav: kõik vajalikud jaotised (isiklik konto, ülesannete loend, täitmise ajalugu) on hõlpsasti juurdepääsetavad. Interaktiivsete funktsioonide hulgast saab kasutaja valida ühe kahest liidese vormingust ülesande lahendamiseks, teiste kasutajatega vestlemiseks, konkreetse ülesande arutamiseks või ülesande testitulemuste vaatamiseks. Uutele kasutajatele aga põhilisi kasutusjuhiseid ei pakuta.

Mis puutub ülesannete täitmisesse, siis selleks saab kasutada erinevate andmebaasisüsteemide (MySQL, DB2, Oracle, MS SQL Server) andmebaase. Samuti pole saidil vahekaarte võimalike lahenduste ja näpunäidetega. Vahendi autorid

soovitavad kasutajatel ideede jagamiseks kasutada ülesande arutelu vahekaarti. Sellele vahelehele ligipääs, ilma et prooviksite ülesannet ise lahendada, võib aga takistada isiklikku õppimist.

Ülesannete lahenduskatsete jälgimiseks pakutakse vahekaarti "Kinnitused", kust saab vaadata lõpetatud katsete loendit koos koodi üksikasjaliku analüüsiga, selle koodi täitmise tulemust ja täitmise olekut. Ka saidi päise all on plokk kasutajate edetabeli ja teenitud punktide arvuga, millel on positiivne mõju kasutaja motivatsioonile SQL-i õppida.

Mis puutub ülesannete otsimisse, siis see toimub mitme tingimuse järgi: täitmise olek, keerukus ja oskuse tase. Saadaval on ainult erinevad andmete otsimise ülesanded. Muud tüüpi lausete kirjutamist nõudvaid ülesandeid ei toetata.

Pakutavate SQL-i ülesannete hulk on väike (umbes 50+ ülesannet). Kui probleemi lahendamise katse ebaõnnestub, siis kahjuks ei pakuta kirjeldust selle kohta, mis valesti läks. Kuvatakse ainult fraas "Vale vastus" ja oodatav tulemus. See platvorm ei võimalda õpetajatel ise SQL-i ülesandeid luua. Selles keskkonnas saab SQLi harjutada tasuta.

3.1.2 LeetCode

LeetCode on veebiteenus, kus õppida programmeerimist ja valmistuda programmeerimist puudutavateks töölevõtuvestlusteks. [7] Sellel on nii tasuline kui tasuta osa. SQL on seal üks keel mitmest, mida saab õppida ja harjutada.

Sait on minimalistlikus stiilis pehmete värvidega, et luua üldine visuaalne mulje. Ülesannete lahendamiseks mõeldud andmetega tabelite kujundamine toimub aga tekstivormingus (vt Joonis 40, Lisa 10), mis võib jätta mulje ebaprofessionaalsusest. Saidil navigeerimine võib mõnikord olla keeruline, kuna kasutajad peavad vajalike funktsioonide leidmiseks liikuma läbi mitme jaotise, nagu "Explore", "Problems" ja "SQL". Küll aga pakutakse juhiseid ja näpunäiteid uutele kasutajatele, mis lihtsustavad probleemide lahendamise lehe struktuuri ja iga ploki funktsionaalsuse mõistmist.

Sait laaditakse kiiresti, mis kiirendab oluliselt SQL ülesannete lahendamist. Kasutajate SQL-i õppimise edenemise jälgimiseks pakutakse tööriistu nii oma lahenduste vaatamise lehel kui ka isiklikul kontol, kus on saadaval juhtpaneel erineva keerukusega ülesannete täitmise protsendiga. Interaktiivsust ja tagasisidet pakutakse avatud vestluse kaudu teiste

saidi kasutajatega, mis võimaldab arutada lahendusi, jätta kommentaare, hinnata ja isegi jagada ülesandeid sotsiaalvõrgustikes. Lisaks on võimalik taimerit seadistada, peatada ja käivitada. Kõik ülaltoodud interaktiivsed elemendid motiveerivad kasutajaid SQL-i õppimist jätkama.

Saidil on ka jaotis "Editorial", kus kasutajad saavad tutvuda erinevate probleemide lahendamise lähenemisviisidega ja uurida teoreetilist materjali. Probleemide võimalikke lahendusi saab vaadata "lahenduste" vahekaardi kaudu, kus kasutajad saavad oma ideid jagada. Siiski puudub lai ülesannete filter ja ülesandeid näidatakse keerukust ja nime kasutades. Süsteem toetab ülesandeid, mille lahendamiseks peab kirjutama SELECT lauseid. Muud tüüpi lauseid, nagu näiteks INSERT, UPDATE ja DELETE, ei toetata. Lisaks on sait saadaval ainult inglise keeles ja toetab ainult MySQL andmebaasisüsteemi.

SQL-ülesannete loetelu on üsna piiratud (umbes 50 ülesannet). Kui katse ülesannet lahendada ebaõnnestub, siis jääb lahendajal kahjuks teadmata, milles täpselt viga oli. Kasutajatel kuvatakse lihtsalt teade "Vale vastus" koos saadud ja oodatud tulemustega. Üks huvitav omadus on võimalus lisada oma lõpetatud katsele kommentaar. Kahjuks ei võimalda platvorm õpetajatel ise SQL-ülesandeid luua. Kuigi teenusel on võimalus lisatasu eest tellida täiendavaid funktsioone, on olemasolev funktsionaalsus piisav.

3.1.3 PostgreSQL Exercises

PostgreSQLi SQL dialekti õppimiseks mõeldud sait paistab silma oma minimalistliku ülesehituse ja ühtse disaini poolest [8]. See on üleval veebisaidina, kuid lähtekood on avalik ning seega oleks võimalik sait täiendusteta või täiendustega ka oma serveris tööle panna.

Selle visuaalset muljet iseloomustab lihtsus ja stiili ühtsus. Põhivärvipalett on vastuvõetav valik, mis tagab rahuldava kasutuskogemuse (vt Joonis 41, Lisa 10). Ülesande täitmise lehel on andmebaasi struktuuri illustatsiooniga plokk, mis näitab tabelite vahelisi seoseid. Juurdepääsu tabeli sisule lehel otse siiski ei pakuta. Saidil navigeerimine on lihtne ja mugav tänu võimalikult lihtsale struktuurile. Kasutajaliidese isikupärastamist piirdub aga võimalusega muuta mõne ploki suurust ülesande täitmise lehel.

Tänu lihtsale liidesele ja juhtnuppudele ülesannete lehel, saavad kasutajad aga intuiitiivselt aru, kuidas seda kasutada. Nupul “abi” klõpsamisel avaneb modaalaken näpunäidetega saidi mugavamaks kasutamise kohta. Lehtede laadimise ja saidil toimingute tegemise kiirus on selle lihtsa struktuuri ja keerukate elementide puudumise tõttu suur. Koodi kontrollimine toimub koheselt (vähem kui kahe sekundiga), mis tagab keskkonna tõhusa kasutamise.

Puuduvad tööriistad, mis jälgiksid kasutajate edusamme SQL-i õppimisel. Kasutajad saavad salvestada ainult oma tulemuse ja vaadata valitud kogust tehtud ülesandeid. Võimalus vaadata kõiki tehtud ülesandeid puudub. Saidil pole interaktiivseid elemente, nagu tugifoorumid või vestlus administraatoriga, mis vähendab tagasiside saamise taset. Vihjeid probleemide lahendamiseks esitatakse modaalakna kujul. Samuti on olemas plokk “Vastused ja arutelu”, kus on üksikasjalikul kujul välja toodud ülesande võimalik lahendus koos lahenduspõhimõtte ja alternatiivsete võimaluste kirjeldusega.

Ülesandeid ei saa otsida, kuid sait pakub erinevaid ülesannete kogumeid (Joins and Subqueries, Aggregates jne). Kõik pakutavad ülesanded on andmete otsimise ülesanded, mille lahendamiseks tuleb kirjutada SELECT lause. Sait ei ole mitmekeelne, see on saadaval ainult inglise keeles. Nagu viitab ka saidi nimi, siis toetatakse ainult PostgreSQL andmebaasisüsteemi. Seega on sait lihtne ja mugav ning minimaalse funktsionaalsusega tööriist SQL ülesannete lahendamiseks, mis muudab selle kättesaadavaks paljudele kasutajatele, kuid toob kaasa ka funktsionaalsuse ja isikupärastamise piirangud.

SQL ülesannete valik on suhteliselt väike (umbes 60 ülesannet). Paraku, kui proovida ülesannet lahendada ja see ebaõnnestub, siis ei anta üksikasjalikku selgitust, mis läks valesti. Kasutajale kuvatakse lihtsalt teade „Vale vastus“ ning saadud tulemus. Samas, kui tekkis süntaksi viga või mõni muu tehniline viga (näiteks pöördumine olematu tabeli poole), siis näidatakse kasutajale selle vea olemust. Platvorm ei võimalda õpetajatel ise SQL ülesandeid luua. Sait on täiesti tasuta pakkudes samal ajal kvaliteetseid haridusplatvorme iseloomustavat funktsioonide komplekti.

3.1.4 SQLPad

SQLPad on veebiteenus, kus saab õppida SQL keelt ja valmistuda programmeerimist puudutavateks töölevõtuvestlusteks [9]. See SQL-i õppimisele keskendunud sait on kaasaegne vahend, millel on mitmekesised kujunduselemendid nagu ikoonid, kaasaegsed

värvid ja animatsioonid, mis loovad meeldiva visuaalse kogemuse. Vaatamata erinevate vahekaartide ja jaotiste rohkusele on saidil navigeerimine käesoleva töö autorite hinnangul mugav, võimaldades hõlpsalt leida vajalikke jaotisi ja funktsioone, nagu ülesande kirjeldus, koodiredaktor ja täitmistulemus (vt Joonis 42, Lisa 10).

Kasutajaliidese isikupärastamine piirdub võimalusega muuta leheploki suurust ja valida kujundusteema. Siiski ei pakuta kasutajale intuitiivseid juhiseid ega viipasid, kui nad esimest korda ülesandelehte külastavad. Kogenud kasutajad saavad aga saidi struktuurist aga hõlpsasti aru, kuna kõik vajalik on ühes kohas ja intuitiivselt on selge, mis mille eest vastutab.

Saidil on suur lehtede laadimise ja toimingute sooritamise kiirus, mis tagab keskkonna tõhusa kasutamise. Ülesannete loendit laadimiseks kulub vähem kui kaks sekundit. Isikliku konto alt saab kasutaja jälgida oma edusamme SQL-i õppimisel. Seal on ülesannete loend, filtreerimine täidetud ja lõpetamata ülesannete järgi, samuti edetabel, mis näitab kuu parimaid kasutajaid, nende hinnanguid ja teenitud punktide arvu. See võib avaldada positiivset mõju kasutaja motivatsioonile SQL-i õppimist jätkata.

Saidi interaktiivsuse ja tagasiside andmise aste on tänu erinevatele elementide olemasolule üsna kõrge, nagu iga ülesande korral "FTPR" väli (inimeste protsent, kes lahendasid probleemi esimesel katsel), lingid teoreetilistele artiklitele ja videomaterjalidele, samuti "liivakastid" tegelike andmetega harjutamiseks. Samuti saavad kasutajad esitada küsimusi ja osaleda foorumi aruteludes, pääsedes vajalikule lehele otse ülesande täitmise lehelt.

Ülesannete lahendamise näpunäiteid pole võimalik vaadata, kuid kasutajad saavad vaadata oma varasemaid lahenduskatseid ja osaleda foorumi aruteludes. Ülesannete võimalike lahenduste vaatamine on saadaval ainult lisatasu maksvate kontode puhul. Ülesannete filtreerimine on piiratud tekstiotsingu ja ettevõttepõhise filtreerimisega (millise ettevõtte töointervjuul on seda küsitud). Sait pakub ainult andmete otsimise üleseandeid (lahendamine nõuab SELECT lausete kirjutamist) ja on saadaval ainult inglise keeles. Kasutajad saavad SQL-i ülesannete täitmiseks kasutada nii PostgreSQLi kui ka MySQLi andmebaasi.

Pakutavate ülesannete valik on muljetavaldav (üle 100 ülesande). Ent kui harjutajal tekib probleeme nende lahendamisel ja katse ebaõnnestub, siis jääb tal teadmata, mis täpselt

valesti läks. Kasutajale kuvatakse lihtsalt teade „Vale vastus“ ning tulemus, mis saadi. Platvormil on olemas tasulise tellimuse võimalus, kuid iseõppimiseks pakutakse piisavalt tasuta funktsioone.

3.1.5 Interaktiivne veebiõpik Tartu Ülikooli õppeainele " Sissejuhatus andmebaasidesse"

See projekt oli loodud bakalaureusetöö raames ning kõik allpool järelused on tehtud viidatud lõputöösse kirjutatu põhjal [10]. Tegemist on veebiõpikuga, kus SQL koodi käivitamine on toetav funktsionaalsus. Kuna vahendi lähtekood on avalik, saaks soovija selle oma serverisse muudatusteta või muudatustega üles panna.

Veebilehel on esitatud info mugavas ja funktsionaalses visuaalses vormingus, kus komponendid on paigutatud ühele lehele, muutes neile juurdepääsu mugavaks ja tõhusaks. Samuti saavad kasutajad keskkonda hõlpsasti mobiilseadmetes kasutada. Saidil navigeerimine on loodud selleks, et võimaldada lihtsat juurdepääsu vajalikele jaotistele ja funktsioonidele. Näiteks algab iga leht õppetunni tekstiga, millele järgneb harjutuste osa, mis võimaldab kasutajatel ülesandeid täita ilma teistele lehtedele liikumata.

Samas ei paku rakendus kasutajaliidese kohandamist teema või elementide paigutuse kohandamise näol. Uutele kasutajatele on saadaval intuiitiivsed juhised ja näpunäited, näiteks ülesannete kohal on näited, mis aitavad probleeme lahendada.

Veebiõpiku autori sõnul on lehtede laadimise ja saidil toimingute tegemise kiirus rahuldavad. Siiski ei pakuta keskkonna kasutajatele võimalusi oma edusammude jälgimiseks. Lisaks näib interaktiivsuse ja tagasiside tase olevat ebapiisav, kuna puuduvad tugifoorumid ja vestluse võimalus administraatoriga.

Õpik ei paku võimalust vaadata näpunäiteid ülesannete lahendamiseks, kuid kasutajad saavad vaadata võimalikke ülesannete lahendusi, mis teeb õppeprotsessi lihtsamaks. Kuid samas ei pakuta ülesannete filtreerimist ja ülesannete lahendamine on võimalik ainult teoreetilise materjali läbimise raames. Süsteem toetab andmete otsimise, ridade lisamise, ridade muutmise, ridade kustutamise, tabelite loomise ülesandeid ja teised SQL lausete tüübid. Keskkond on eesti keeles ja kasutada saab ainult PostgreSQL andmebaasisüsteemi.

Pakutavate ülesannete loetelu on platvormil käsitletavate teemade arvu arvestades väike ja neid saab vaadata teemade kaupa. Selle vahendi looja väidab, et kasutaja näeb vigade korral, mis viga põhjustas. Kuigi pole mainitud, kas õpetajatel on võimalus ise ülesandeid luua, võib eeldada, et saab lisada uusi teoreetilisi mooduleid ja nendega seotud ülesandeid, kuid see nõuab programmeerimist. Platvorm on tasuta ning saadaval Tartu Ülikooli kursuse raames või iseseisva installeerimiseks GitHubist.

3.2 Teaduskirjandus

Selles lõputöös käsitletakse SQLi iseseisva harjutamise veebirakenduse loomist, mis on suunatud üliõpilastele SQL-i praktiseerimiseks. Käesolev jaotis käsitleb teemaga seotud teaduskirjandust. Selleks toetutakse neljale artiklile, millest esimene keskendub semantilistele vigadele, teine süntaktilistele vigadele, kolmas SQL lausete automaatsele hindamisele ja neljas SQLi õpetamisele. Tegemist on väga väikese osaga teemakohasest kirjandusest, kuid see tõstab esile erinevaid uurimissuundi.

Esimene teadusartikkel analüüsib semantilisi vigu, mida tudengid teevad SQL SELECT lausete kirjutamisel. Uuringu tulemusena selgub, et enamik semantilisi vigu on „ärajätmise“ tüüpi, mis tähendab, et üliõpilased jätaavad vajalikud osad päringust välja. Need vead esinevad eriti lausetes, mis nõuavad tabelite ühendamist (JOIN operatsiooni), alampäringut või ridade grupeerimist (GROUP BY klauslit). Uuring rõhutab vajadust õpetada üliõpilasi paremini valima sobivaid lausekonstruktsioone ja kasutama päringu koostamisel süstemaatilist lähenemist. Uuringus tuuakse välja, et SQL õpetamisel tuleks rõhku panna tehnikatele, mis aitavad üliõpilastel määrata sobiva lausekonstruktsiooni ning õpetajatel tuleks kasutada selget terminoloogiat, et vähendada semantiliste vigade arvu (A. Ahadi, J. Prior, V. Behbood, R. Lister, 2016) [11].

Teine teadusartikkel keskendub üliõpilaste poolt SQL lausete kirjutamisel tehtud süntaksivigade analüüsile. Uuring esindab esimest suuremahulist kvantitatiivset analüüsi üliõpilaste süntaksivigade kohta erinevat tüüpi SQL lausetes. Uuringus koguti kaheksa aasta jooksul üle 160 000 SQL lause üle 2000-lt üliõpilaselt. Leiti, et üldiselt teevad üliõpilased rohkem süntaksivigu kui semantilisi vigu. Süntaksiviga on sageli põhjus, miks üliõpilane loobub ülesande lahendamisest. Kõige levinumad vead on süntaksivead erinevates SELECT lause osades ja grupeerimisvead. Uuring rõhutab, et SQL lausete õpetamisel tuleks pöörata suuremat tähelepanu süntaksile ja süntaksivigadele, kuna need

vead takistavad üliõpilastel semantilistest vigadest aru saamist (A. Ahadi, J. Prior, V. Behbood, A. Vihavainen, R. Lister, 2016) [12].

Kolmas teadusartikkel tutvustab uut mudelit SQL lausete automaatseks hindamiseks, mis peaks aitama nii üliõpilasi kui ka õpetajaid. Mudel võimaldab üliõpilastel saada kiiremini tagasisidet ja hinnanguid tehtud vigade kohta, mis on oluline nende õppimisprotsessis. Mudel keskendub üliõpilaste lahenduste osalisele hindamisele ja annab õpetajatele tööriista hindamisprotseduuride täpselt konfigureerimiseks. Uuring näitas, et automaatse hindamissüsteemi antud hinnangud ei erine oluliselt õpetajate antud hinnangutest, mis viitab sellele, et süsteemi kasutamine ülikoolikeskkonnas on paljulubav. Siiski on süsteemil praegu piirangud, kuna see toetab ainult lihtsaid SELECT lauseid ilma JOINide, GROUP BY ja HAVING klauslite, kokkuvõttefunktsioonide ja alampäringuteta. Samuti avastas süsteem õpetajate käsitsihindamises esinenud väikseid, kuid olemasolevaid ebakõlasid, mis viitavad sellele, et automaatne süsteem võib parandada hindamisprotsessi järjepidevust ja täpsust (M. Fabijanić, G. Đambić, B. Fulanović, 2020) [13].

Viimane artikkel, käsitleb SQLi laialdast kasutust ning selle õpetamist kõrgkoolide programmeerimiskursustel. Viimastel aastatel on välja töötatud mitmeid veebipõhiseid tööriistu, mis aitavad SQL-i õppida ja hinnata. Artiklis arutatakse erinevate vigade üle, mida õpilased SQL-i päringute kirjutamisel teevad, ja kuidas õige hindamine ning õigeaegne tagasiside aitavad õpilastel oma SQL-i teadmisi parandada. Peamised punktid hõlmavad SQL-i olulisust programmeerimiskursustel, hindamise ja tagasiside tähtsust õppimisprotsessis, automaatsete hindamissüsteemide vajadust õpetajatele ning õpilaste tavapärase vigade kategoriseerimist sünonüümide, süntaksi ja semantilisteks vigadeks. Käsitletakse üksikasjalikult automaatsete hindamistööriistade, nagu SQL Tester, SQL-Statement Grader, SQL Knowledge Tester, SQLify, Active SQL ja SQLator tugevusi ja nõrkusi, tuues esile, et paljud tööriistad ei paku piisavalt tagasisidet. Artiklis rõhutatakse vajadust uue süsteemi järele, mis suudaks ületada olemasolevate tööriistade piirangud ja pakkuda põhjalikumalt ning personaliseeritud tagasisidet, aidates õpilastel SQL-i tõhusamalt õppida ja parandada nende üldist õppimiskogemust. (S. Nayak, R. Agarwal, S. Kumar Khatri, 2022) [14].

Kokkuvõttes näitavad need uuringud, et SQL õpetamisel on oluline keskenduda nii süntaktilistele kui ka semantilistele vigadele. Süntaksivigade rohkus ja nende mõju

üliõpilaste edasisele õppimisele viitab vajadusele parandada süntaksi reeglite õpetamist. Samal ajal on semantiliste vigade analüüs ja nende parandamine samuti oluline, et üliõpilased suudaksid koostada korrektseid ja tõhusaid SQL lauseid. Automaatse hindamissüsteemi kasutuselevõtt võib oluliselt kiirendada ja parandada tagasiside andmise protsessi, aidates seeläbi üliõpilastel oma vigadest õppida ja parandada oma oskusi SQL lausete koostamisel. Loodud veebirakendus peaks sisaldama suurt hulka eriilmelisi praktilisi ülesandeid, et toetada õppimist ja vähendada selle kaudu vigade arvu. Süsteemi poolt lahenduskatsete logimine võimaldaks tulevikus analüüsida õppijate tehtud vigu.

4 Töö tulemused

Selles peatükis esitatakse käesoleva töö põhitulemused. Töö autorid said sisendina süsteemi metaandmete andmebaasi loomise laused, sh tabelite ja funktsioonide loomise laused, kasutajate loomise ja õiguste jagamise laused ning klassifikaatorite väärtuste ja testülesannete registreerimise laused. Loodud andmebaasis oli 28 baastabelit, üks vaade ja kaheksa funktsiooni. Näitena loodud baastabelite kohta esitatakse lisas 4 ülesannete registri andmebaasi disaini kirjeldavad skeemid. Funktsioonideks oli kasutaja tuvastamise funktsioon ja funktsioonid harjutaja poolt sisendiks antud SQL lause kontrollimiseks enne käivitamist (kas lause vastab ülesande tüübile, kas lause ei sisalda keelatud mustreid, kas lause sisaldab nõutud mustreid), lahenduskatse registreerimiseks ja lahendamise ajaloo kustutamiseks. Mustrid on regulaaravaldise mustrid ja need mustrid on registreeritud metaandmete andmebaasis nii, et administraator saab neid ka igal ajal muuta. Samuti said autorid sisendiks testülesannete andmebaasi loomise laused. Lisaks oli töö autoritele sisendiks ülesannete kontrollimise algoritmi kirjeldus. Autorite ülesandeks oli luua toimiv ja hästikasutatav kasutajaliides harjutajatele ning selle käigus realiseerida ka ülesannete kontrollimise algoritm.

4.1 Tarkvara funktsionaalsus

Selles jaotises selgitatakse loodud tarkvara erinevaid funktsionaalsuseid. Järgnev ülevaade arvestab juba muudatustega, mida tehti esimese valminud versiooni kohta kasutajatelt saadud tagasiside alusel.

4.1.1 Autoriseerimine Mauruse süsteemi kaudu

Veebilehele sisselogimine nõuab autentimist, kasutades andmebaaside õppeainetes kasutusel oleva Mauruse süsteemi jaoks kehtivat kasutajanime ja parooli. Selleks luuakse ühendus väliseenuse andmebaasiga ja käivitatakse seal loodud funktsioon, millele antakse ette kasutajanimi ja parool. Kui kasutajal on õigus SQLi harjutamise süsteemi siseneda, siis tagastab funktsioon kasutaja ID, mida kasutatakse seejärel kasutaja identifitseerimiseks harjutamise süsteemis (see kirjutatakse ülesannete lahenduskatsete ja kasutaja tagasiside juurde). Pärast autentimist antakse kasutajale juurdepääs erinevatele lehtedele, nagu ülesannete loend, ülesannete lahenduse leht, statistika, ajalugu ja ülesannete täitmise ajalugu. Väliseenuses loodud funktsioon oli töö autoritele ette antud.

4.1.2 Keele valik

Saidi avalehel on keelevelik ja selle valik on saadaval ainult pealehel. Hetkel pakutakse kahte keelevelikut: eesti ja inglise keel. Saidi administraatoril on võimalus lihtsasti lisada uusi keeli, registreerides andmebaasis klassifikaatorite väärtuste tõlked uude keelde ja lisades vastava kasutajaliidese tõlkefaili rakenduses spetsiaalsesse "locales" kataloogi.

Keele valik mõjutab süsteemi kasutajaliidest, sealhulgas süsteemi vastuseid ülesannete lahenduskatsetele ja staatilise teksti tõlkimist saidil. Lisaks mõjutab see ka saadaolevaid ülesandeid, kuna need sõltuvad valitud keelest. See mehhanism võimaldab saidi kasutajatel suhelda platvormiga keeles, milles neile kõige mugavam on. Administraator saab luua erinevates keeltes õppijatele erinevaid ülesandeid. Iga ülesanne ning selle vihjed ja kirjeldused on seotud ühe kindla keelega. Valides pealehelt keele näeb kasutaja süsteemi sisenedes selle keele jaoks loodud ülesandeid.

4.1.3 Ülesande otsimine

Ülesannete loendi leht pakub laia valikut funktsioone ülesannete hõlpsaks otsimiseks, filtreerimiseks ja sortimiseks. Kasutajal on võimalus otsida ülesannet erinevate kriteeriumide järgi, kasutades erinevaid filtreerimismeetodeid.

Esimene saadaolev filtreerimistüüp on ülesande nimeotsing, kus kasutaja saab konkreetse ülesande otsimiseks sisestada tekstiväärtuse. Täiendav filtreerimine on saadaval lause tüübi järgi, mis sisaldab valikuid SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE ja DROP TABLE. Need on erinevad lausete tüübid, mida käesolev süsteem toetab.

Lisaks on võimalik filtreerida ülesande raskusastme järgi, kus kasutaja saab valida: Algtase, Edasijõudnutele ja Eksperttase. Veel üheks filtreerimise kriteeriumiks on andmebaas, mille põhjal saab ülesandeid lahendada. Iga ülesanne on seotud kindla andmebaasiga.

Samuti on saadaval filtreerimine ülesande lahendamise seisundi järgi, kus kasutaja saab valida Lahendatud, Lahendamata ja Proovimata. Ülesande lahendamise seisundi määramine toimub järgmistel põhimõtetel. Kui kasutaja ülesande lahendamise katsete hulgas on vähemalt üks õige vastuse andnud katse (see ei pea olema kõige viimane katse), siis on ülesande lahendamise seisundiks "Lahendatud". Kui kasutaja on proovinud

ülesannet lahendada (katseid on üks või enam), kuid pole ühtegi edukat katse, siis on ülesande lahendamise seisundiks "Lahendamata". Kui kasutaja pole kordagi proovinud ülesannet lahendada, siis on ülesande lahendamise seisund "Proovimata". Lisaks on kasutajal juurdepääs plokis asuvate ülesannete filtreerimisele kogumite järgi (vt Joonis 10, Lisa 5).

Eraldi plokis on ülesannete kogumite loend, kus on näidatud kogumi nimi, kogumisse kuuluvate täidetud ülesannete arv ja kogumi ülesannete koguarv. Kogumi otsimise kiirendamiseks on saadaval selle nime järgi filtreerimine.

Kasutajate mugavuse huvides on võimalik ülesandeid sorteerida erinevate kriteeriumide järgi nagu nimi, andmebaasi nimi, lause tüüp, keerukus ja ülesande lahendamise seisund. Kui valitakse kogum, siis kuvatakse selle kirjeldus ülesannete loendi kohal ja lisaks esitatakse ülesannete tabelis veerg, mis sisaldab ülesande järjekorranumbrit kogumis. Eeldatakse, et ülesanded tuleks lahendada selles järjekorras (väikseimast suurimani), kuid seda järjekorda süsteem kuidagi kohustuslikuks ei muuda. Süsteem annab võimaluse ülesannete tabelit ka selle veeru järgi sorteerida.

Lisaks pakub ülesannete loendi leht funktsiooni lahendamiseks juhusliku ülesande valimiseks, mis võib olla kasulik kasutajatele, kes soovivad lasta süsteemil ennast korraks proovile panna. See mehhanism lisab loodud platvormile interaktiivsust, pakkudes lisavõimalust kasutajakogemuse rikastamiseks.

4.1.4 Ülesande lahendamine

Ülesande lahendamise leht pakub laia valikut tööriistu ja funktsioone, et suurendada kasutajate mugavust ja võimaldada neil tõhusalt töötada.

Esimene ja peamine funktsionaalsus on võimalus ülesannet ise lahendada. Kasutajal on juurdepääs kõikidele ülesande andmetele, sealhulgas selle kirjeldusele, pealkirjale, raskusastmele, lause tüübile, andmebaasile ja lahendamise seisundile. See võimaldab kasutajatel ülesandest aru saada.

Lisaks on võimalik uurida andmebaasi struktuuri, mille põhjal tuleb ülesanne lahendada. See hõlmab andmebaasi kirjelduse lugemist ja andmebaasi struktuuri illustreeriva joonise vaatamist. Andmebaasi baastabelite sisu täpsemaks uurimiseks on võimalik vaadata ka iga üksiku tabeli sisu. Iga süsteemis olev ülesanne põhineb kindlal andmebaasil. Süsteemi

administraatoril on võimalus kerge vaevaga lisada uusi andmebaase, mille põhjal saab ülesandeid lahendada.

Teine oluline ja väga kasulik funktsioon on kasutaja võimalus tutvuda ülesande lahendamise eeldatava tulemusega, mis annab aimu, milline peaks olema tema lahenduse tulemus, olenevalt ülesande tüübist. Selleks on nupp "Vaata oodatud tulemust", mis avab modaalakna päringu tulemuse tabeliga.

Kasutajad saavad ülesande lahendamiseks vaadata vihjeid. Administraator saab konfiguratsioonifailis määrata vihjete vaatamise eelduseks oleva katsete minimaalse arvu. Samuti võib kasutaja vaadata vihjeid pärast oma õige vastuse saamist. Kasutajatel on võimalus vaadata vihjeid kahel viisil: ükskhaaval või kõiki korraga.

Samuti on võimalik vaadata ülesande võimalikke lahendusi. Administraator saab konfiguratsioonifailis määrata õigete lahenduste vaatamise eelduseks oleva katsete minimaalse arvu. Kasutajatel on võimalus vaadata võimalikke lahendusi kahel viisil: ükskhaaval või kõik korraga. Samuti saab administraator soovi korral täielikult keelata juurdepääsu saadaolevatele lahendustele, väärtustades konfiguratsioonifailis ühte juhtparameetrit sobival viisil. Sel juhul saab võimalikke lahendusi vaadata vaid siis, kui kasutajal on katsete hulgas juba vähemalt üks edukas katse.

Saidi administraatorile tagasiside andmiseks on olemas sõnumite saatmise funktsionaalsus, kuhu kasutaja saab konkreetse ülesande kohta kommentaari jätta. Administraatoril on omakorda võimalus kasutaja sõnumile vastata (selle funktsionaalsuse realiseerimine ei kuulunud töö skooopi, kuid administraatori töölaarakendus on olemas). Sel juhul näeb kasutaja vastavas modaalaknas nii oma sõnumeid kui ka administraatori vastuseid.

Redaktoris oma SQL-koodi kirjutamine võimaldab kasutajatel probleemide lahendamiseks SQL lauseid sisestada. Redaktoril on järgmised omadused.

Esiteks tõstab süntaksi esiletõstmise automaatselt esile SQL-koodi põhielemendid, muutes kasutajale koodi loetavamaks ja arusaadavamaks. Samuti annab füüsiliste koodiridade arvu kuvamine kasutajale infot, mis võib olla talle kasulik saadud lahenduse efektiivsuse ja loetavuse hindamisel.

Teine kasulik funktsioon on koodi jätkamise valikute vaatamine, mille nägemiseks tuleb vajutada Ctrl+Space. Need annavad kasutajale võimaluse valida SQL-i võtmesõnu (näiteks SELECT, AND, OR, TABLE, ORDER) ja tabelite nimesid. See vähendab koodi kirjutamisele kuluvat aega ja aitab kasutajal vältida vigu.

Ülesande täitmiseks kuluva aja jälgimiseks pakub süsteem taimerit, mis käivitub ülesande avamisel ja peatub pärast ülesande õigesti lahendamist. See lehel olev interaktiivne element loodi eesmärgiga suurendada kasutaja keskendumist ülesande täitmisele ja suurendada distsipliini.

Ülesande lahendamise leht on jagatud kahte plokki: vasakpoolne, mis sisaldab ülesande kirjeldust, andmebaasi struktuuri, vihjeid ja võimalikke lahendusi ning parempoolne, kus asub ülesande lahendamise koodiredaktor ja kuvatakse kasutaja koodi käivitamise tulemus. Nende kahe ploki vahele on pandud liugur, mida tõmmates saab lehe vasaku/parema osa laiust suurendada/vähendada, võimaldades kasutajal vajadusel teatud saidi osale rohkem tähelepanu pöörata. Näiteks, kui kasutaja on ülesande kirjeldusega piisavalt kursis, siis saab ta koodi kirjutamise osa suuremaks venitada.

Lõpuks saavad kasutajad lasta süsteemil kontrollida oma ülesande lahendust. Pärast lahenduse kontrollimiseks esitamist saab kasutaja ülesande täitmine kas seisundi "Lahendatud" või "Lahendamata". SELECT, INSERT, UPDATE, DELETE või MERGE lause õnnestumise korral saab kasutaja vaadata lause tulemust tabeli kujul. Andmete vaatamise hõlbustamiseks on ette nähtud tulemuse lehekülgedeks jagamine. INSERT, UPDATE, DELETE või MERGE lause korral näeb kasutaja muudetud tabelis olevaid andmeid sellisena, nagu need olid peale muutmist. SELECT, INSERT, UPDATE, DELETE või MERGE lause korral saab kasutaja võrrelda enda tulemust oodatava tulemusega.

Kui aga ülesanne lahendati valesti, siis saab kasutaja veateate, mis kirjeldab vea olemust. Teade võib viidata erinevatele probleemidele, näiteks lause ei vasta ülesandes määratud lause tüübile (nt DELETE lause SELECT lause asemel) või puudub nõutud sorteerimine. Lisaks, kui lause täitmisel ilmneb tõrge, siis kuvatakse probleemi kohta lisateabe saamiseks PHP poolt antud vead ja andmebaasisüsteemi tagastatud vead (vt Tabel 4, Lisa 3, mis sisaldab võimalike vigade täielikku loendit).

4.1.5 Ülesannete lahendamise ajalugu

Ülesannete lahendamise ajaloo lehel on mitmeid põhifunktsioone, mis pakuvad kasutajatele hõlpsat juurdepääsu lõpetatud ülesannetele ja nende tulemustele.

Kasutajad saavad vaadata kõigi ülesannete lahendamise ajalugu ja seega ka oma edenemist. Vajaliku teabe leidmise hõlbustamiseks saab kasutada filtreerimist. Kasutajad saavad ülesannete lahenduskatseid filtreerida algus- ja lõppaja ning lahendamise seisundi (lahendatud, lahendamata) järgi.

Lahenduskatsete loendi tabelist on võimalus sorteerida katseid erinevate kriteeriumide järgi, nagu aeg, ülesande nimi, andmebaas, milles ülesanne täideti, lause tüüp, keerukus ja lahendamise seisund. See pakub lahenduskatsete otsimisel ja analüüsimisel mugavust. Lahenduskatsete tabeli kohal kuvatakse leitud katsete koguarv, mis võimaldab kasutajatel hinnata oma töö ja tegevuse kogumahtu.

Vajadusel saavad kasutajad vaadata üksikasjaliku infot katse kohta, klõpsates vastaval katsel. See võimaldab tulemusi üksikasjalikumalt analüüsida ja annab täiendavat kontekstipõhist teavet. Kui neil on õigus näha õiget vastust, siis saavad nad võrrelda oma lahenduse ja õige lahenduse koodi.

Sellel lehel on ka võimalus kustutada oma katsete ajalugu. Lisaks on saidi administraatoril võimalus konfiguratsioonifaili kaudu lubada või peita ülesannete ajaloo kustutamine. See võimaldab blokeerida kasutajatel saidi ülesannete täitmise ajaloo tühjendamise – näiteks ajaperioodil kui keskkonnas tehtud katseid kasutatakse hindamiseks. Oluline on märkida, et tegelikult ei kustutata andmebaasist andmeid lõpetatud katsete kohta, vaid kustutatakse ainult kasutaja seos nende katsetega e andmed anonümiseeritakse. Niimoodi säilib võimalus uurida ülesannete lahendamisel esinevaid probleeme.

4.1.6 Ülesande lahendamise ajalugu

Konkreetse ülesande täitmise ajaloo leht annab kasutajatele võimaluse üksikasjalikult analüüsida kõiki enda poolt tehtud antud ülesande lahendamise katseid. Esiteks on oluline märkida, et kasutajatel on võimalus vaadata konkreetse ülesande lahendamise ajalugu, mis võimaldab neil saada kogu vajalikku teavet antud ülesande lahendamise katsete kohta. Esimene kasulik omadus on võimalus vaadata antud ülesande lahendamiseks

tehtud katsete koguarvu, mis võimaldab kasutajatel hinnata oma edusamme ja aktiivsuse taset.

Järgmisena saavad kasutajad vaadata viimast ülesande lahendamise katset, mis annab infot katse kontrollimise tulemuse ning lõpuleviimise aja kohta ja esitab ka hästi loetavale kujule vormindatud kasutaja SQL-koodi. Lisaks, kui lahenduse kohta on süsteemi poolt antud tagasisidet, siis kuvatakse see samuti sellel lehel, mis võimaldab kasutajatel saada täiendavaid kommentaare ja vihjeid oma koodi täiustamiseks.

Täpsema analüüsi jaoks on kasutajatel võimalus vaadata loendit kõigist valitud ülesande lahendamise katsetest. Iga katse puhul kuvatakse selle tegemise aeg ja ülesande lahendamise seisund (Lahendatud/Lahendamata). Vajadusel saavad kasutajad valida konkreetse katse, klõpsates linki „Vaata rohkem“, et näha üksikasjalikumat teavet kasutaja käivitatud SQL-koodi kohta ja võimaluse korral tagasisidet selle koodi kohta. Kui neil on õigus näha õiget vastust, siis saavad nad võrrelda oma lahenduse ja õige lahenduse koodi. Samal lehel on ka nupp ülesande lahendamise lehele minemiseks, et ülesande andmeid täpsemalt uurida.

4.1.7 Statistika

Statistika lehel saavad kasutajad tervikliku ülevaate oma tegevusest ja edust süsteemis ülesannete lahendamisel. Statistika leht on jagatud semantilisteks plokkideks, millest igaüks vastutab saidi ülesande täitmise konkreetse aspekti analüüsimise eest.

Esimene statistikaplokk on globaalne statistika, kus kasutajad saavad vaadata tehtud katsete koguarvu, edukate ja ebaõnnestunud katsete arvu, edukate ja ebaõnnestunud katsete protsenti katsete koguarvust ning kokku katsetele kulutatud aega.

Sellele järgneb plokk statistikaga ülesannete raskusastmete kohta. Iga raskusastme (Algtase, Edasijõudnutele, Eksperttase) kohta kuvatakse ülesannete koguarv, kasutaja poolt lahendatud ülesannete arv ja kõigi selle raskusastmega ülesannete täitmise protsent. Juhul kui andmebaasi lisada mõni uus raskusaste, siis on kasutajal võimalus ülesannete loendit selle uue raskusastme järgi filtreerida, samuti saab administrator määrata oma tulevastele või olemasolevatele ülesannetele uue raskusastme.

Järgnevalt tuleb suur sektordiagrammidega plokk, mis annab teavet iga lausetüübi ülesannete täitmise kohta. Iga lausetüübi (SELECT, INSERT, UPDATE, DELETE,

MERGE, CREATE TABLE, ALTER TABLE ja DROP TABLE) puhul näidatakse täidetud ülesannete protsenti ja tegelikku sooritatud ülesannete arvu.

Saidi lõpus on kaks statistikaplokki: andmebaaside statistika ja ülesannete kogumite statistika. Esimene plokk sisaldab andmeid kasutajate lahendatud ülesannete arvu ja kõigi ülesannete täitmise protsendi kohta iga andmebaasi kohta, millel põhineb mõni ülesanne. Ülesannete kogumite plokkis kuvatakse iga kogumi kohta ülesannete koguarv, kasutaja lahendatud ülesannete arv ja iga kogumi kõigi ülesannete täitmise protsent. Kuvatakse nii aktiivseid kui ka mitteaktiivseid kogumeid.

Nii annab statistikaleht kasutajatele mitmekesisist infot nende tegevustest ja edusammudest ülesannete lahendamisel, võimaldades teadlikumalt oma õppimist juhtida ja oskuseid täiendada.

4.2 Mittefunktsionaalsed omadused

Selles jaotises selgitatakse tarkvara omadusi, mis pole otseselt seotud funktsionaalsusega.

4.2.1 Esitus

Loodud keskkonnas pööratakse tähelepanu jõudluse optimeerimisele. Lehed laaditakse kiiresti, kasutades selleks mitmeid meetodeid.

Esiteks kasutatakse andmete laadimiseks serverist AJAX-i päringuid, mis võimaldab kasutajal saada teavet ilma lehte täielikult uuesti laadimata. Näiteks ülesannete nimekirja laadimisel saadab süsteem serverile päringu ja uuendab dünaamiliselt lehe sisu, mis vähendab kasutaja ooteaega ja võiks suurendada kasutajate rahulolu platvormi kasutamisel.

Lisaks kasutatakse lehe laadimisaja minimeerimiseks meetodit, kus laaditakse ainult praeguse lehe jaoks vajalikud skriptid ja stiilid. See on eriti oluline lehtede puhul, mis nõuavad mahukate teekide (nt jQuery või Bootstrap) laadimist. See lähenemisviis aitab vähendada edastatavate andmete hulka ja kiirendada lehtede laadimist.

Serveri koormuse vähendamiseks ja lehe laadimisaja lühendamiseks kasutame rasterpiltide asemel ka SVG-vormingut. See kehtib eriti ikoonide ja muude graafiliste

elementide kohta, kus SVG-vorming tagab väikese failisuurusega kõrge pildikvaliteedi, mis parandab lehe laadimise kiirust.

Hoolimata asjaolust, et iga kord, kui SQL lauset kontrollitakse, luuakse kasutaja andmebaas uuesti, annab server kiiresti vastuse. Selline lähenemine võimaldab süsteemil taotlusi kiiresti töödelda ja neile kiiresti vastata, hoolimata kasutajate andmebaaside uuesti loomisest.

4.2.2 Turvalisus

Veebiplatvormi turvalisus on üks olulisemaid aspekte, millele arenduse käigus erilist tähelepanu pöörata. Järgnevalt esitletakse põhilisi turvameetmeid, mida saidil rakendatakse andmekaitse tagamiseks ja ohtude ennetamiseks.

PHP sisseehitatud seansside kasutamine tagab kasutaja seansi teabe turvalise salvestamise. Sinna salvestatavad andmed on kasutaja ID, seansi eluiga, CSRF-märgid ja kasutajaliidese keel. Lisaks salvestatakse tundlikud andmed, näiteks andmebaasiühenduse teave, eraldi konfiguratsioonifailis, mis asub väljaspool veebiserveri haaret, vähendades seega teabelekke ohtu.

CSRF-i lubade kasutamine saidi iga päringu jaoks aitab kaitsta saidiüleste päringu võltsimise rünnakute eest. Seansi eluiga on piiratud ühe päevaga, mis aitab vähendada riske, mis on seotud aktiivse seansi pikaajalise salvestamisega.

Turvalisuse oluline aspekt on kõigi serveri vigade (warning, exception, Notice, Fatal Error, Unknown) logimine. Veateadete ilmnmisel salvestatakse need automaatselt vastavatesse logifailidesse, mis võimaldab administraatoril neid vajadusel analüüsida.

Juurdepääsu piiramiseks saidi erinevatele osadele on konfigureeritud .htaccess-fail. Juurdepääs on avatud ainult teatud olemitele, näiteks index.php fail ja teatud kataloogidele (db_pictures, frontend (välja arvatud kaust leheküljed)), samas kui juurdepääs teistele osadele: backend, frontend, locales, logs ja systemClasses on täielikult blokeeritud (kui proovida minna nendes osades mistahes faili, siis see annab vea 403). See vähendab tundlikele andmetele volitamata juurdepääsu tõenäosust.

Marsruutimine saidil toimub läbi index.php faili, mis võimaldab tõhusalt kontrollida juurdepääsu erinevatele lehtedele ja ressurssidele. Kõik katsed teha veebilehe kaudu

volitamata tegevust, sealhulgas katsed pääseda ligi ligipääsmatutele ressurssidele, on õigesti konfigureeritud marsruutimise tõttu blokeeritud.

HTTPS-protokolli kasutamine tagab kliendi ja serveri vahel edastatavate andmete krüptimise, mis takistab ründajate poolt teabe pealtkuulamise ja asendamise võimalust.

Arendajatele olid ette antud andmebaasi funktsioonid SQL lause kontrollimiseks enne käivitamist. Nende kontrollide hulka kuulub lause kontrollimine erinevate regulaaravaldiste mustritega, mida andmebaasi administraator saab vajadusel muuta ja mis tagavad selle, et andmebaasis ei saa käivitada koodi mis teeb mitte-ettenähtud tegevusi. Niimoodi kaitstakse keskkonda muuhulgas SQL süstimise rünnete suhtes.

Lähtutakse põhimõttest, et rakendus suhtleb andmebaasiga kui piiratud õigustega andmebaasi kasutaja (st mitte kui ülikasutaja), kellele on antud õiguseid minimaalsuse põhimõttest lähtuvalt – nii palju kui vaja ning nii vähe kui võimalik. Andmebaasi kasutaja, mis küsib andmeid metaandmete andmebaasist ning andmebaasi kasutaja, mis käivitab kontrollimiseks kasutaja koodi tema andmebaasis on erinevad ja erineva õiguste hulgaga.

Seega kõigi nende turvameetmete rakendamine koos tagab veebiplatvormi kaitse ja kasutajaandmete privaatsuse.

4.2.3 Töökindlus

Veebiplatvormi töökindlus mängib olulist rolli stabiilse töö tagamisel ja võimalike rikete ennetamisel. Järgnevalt käsitletakse saidil rakendatud peamisi meetmeid, et tagada kõrge töökindlus.

Alustuseks töödeldakse saidil esinevaid vigu globaalse proovivõtuploki abil. See võimaldab erandeid tabada ja käsitleda, vältides rakenduste katkestusi ja võimaldades vigu sujuvalt käsitleda. Selline lähenemine vähendab oluliselt tundmatute vigade esinemise riski saidil, mis aitab suurendada selle töökindlust.

Rikete või ootamatute olukordade korral saab klient serverilt tagasisidet. Näiteks ootamatu vea ilmnemisel teavitatakse kasutajat probleemist, mis võimaldab tal toimuvast teadlik olla ja vastavaid meetmeid rakendada. See aitab parandada kasutajakogemust ja suurendada usaldust platvormi vastu.

Erilist tähelepanu pööratakse kasutajate andmebaasiga töötamisel andmete terviklikkuse tagamisele. Kui ilmneb andmebaasi tõrge, siis tühistatakse kõik muudatused andmebaasi tehingu tagasipööramise mehhanismi abil. Selline lähenemine tagab, et probleemi ilmnmisel jäävad kasutaja andmed puutumata, mis on platvormi töökindluse tagamisel võtmetähtsusega aspekt.

Seega võimaldab nende meetmete kasutamine tagada veebiplatvormi töökindluse, minimeerides rikete riski ja tagades süsteemi stabiilse töö.

4.2.4 Hooldatavus

Veebiplatvormi hooldatavus mängib olulist rolli selle tõhusa toimimise ja administraatori kasutajasõbralikkuse tagamisel. Järgnevalt käsitletakse saidil rakendatud hooldatavuse peamisi aspekte.

Käesoleva töö ülesannete hulka ei kuulunud administraatori haldusliidese loomine. Kuna ülesannete, kogumite, vihjete, võimalike lahenduste, lahenduskatsete ja muu säärane info salvestatakse andmebaasis, siis saab administraator luua endale ise andmebaasirakenduse või kasutada mõnda andmebaaside haldamiseks mõeldud programmi.

Faili *config.php* kaudu on administraatoril võimalus hallata parameetrite väärtuseid, mis juhivad süsteemi tööd. Allpool on oluliste juhtparameetrite üksikasjalikum kirjeldus.

Parameetri *system.min_count_to_see_solutions* väärtus määrab minimaalse katsete arvu, mis on vajalik, et kasutaja saaks vaadata ülesannete õigeid lahendusi. Kui selle parameetri väärtus on suurem kui null, näeb kasutaja lahendusi alles pärast määratud arvu katsete tegemist. See soodustab kasutajate õppimist ja arengut, kuna nad on sunnitud enne õigete lahenduste juurde pääsemist ülesandeid ise lahendama. Kui väärtus on väiksem kui null, siis näeb kasutaja õigeid lahendusi ainult siis, kui ta ise selle ülesande vähemalt korra õigesti ära lahendab. Kui väärtus on 0, siis näeb kasutaja õigeid vastuseid kohe.

Samamoodi määrab parameetri *system.min_count_to_see_hints* väärtus vihjete nägemiseks vajalike katsete minimaalse arvu. Kasutajad näevad vihjeid alles pärast teatud arvu katseid. See julgustab kasutajaid enne vihjete poole pöördumist proovima probleemi ise lahendada, mis aitab neil õppida ja oma oskusi arendada.

Parameetri *system.can_delete_history* väärtus reguleerib kasutaja võimalust saidil oma lausete käivitamise ajalugu kustutada. Kui väärtus on 0, siis see võtab kasutajalt täitmisajaloo kustutamise võimaluse, samas kui väärtuse 1 määramine annab kasutajale selle võimaluse. See funktsioon annab administraatorile täiendava kontrolli kasutaja tegevuse üle saidil ja nende platvormiga suhtlemise ajaloo haldamise üle.

Nende võimaluste juurutamine suurendab veebiplatvormi ülalpidamise mugavust ja efektiivsust, pakkudes administraatoritele võimalusi saidi funktsionaalsuse haldamiseks.

4.3 Kasutajaliides

Selles jaotises käsitletakse kasutajaliidest. Loodud kasutajaliidese ekraanipildid esitatakse lisas 5.

4.3.1 Värvipalett

Haridusplatvormi värvipaleti väljatöötamisel valiti minimalistlik ja range lähenemine. See on tingitud kasutajate – õpilaste – vajadustest, kes on peamiselt keskendunud probleemide lahendamisele. Oluline oli, et saidi visuaalne kujundus ei tõmbaks nende tähelepanu õppeprotsessilt kõrvale, mistõttu otsustati luua tagasihoidlik värvilahendus.

Platvormil kasutatavad värvid on valitud nii, et need ei tekitaks asjatut visuaalset ärritust. Heledate ja küllastunud toonide asemel valiti mõõdukamad ja neutraalsemad toonid, mis aitab luua saidil rahuliku ja tasakaalustatud atmosfääri. See lähenemine aitab kasutajatel paremini keskenduda ülesannete lahendamisele.

Üldiselt on minimalistlik ja range värvipalett haridusplatvormi kujunduse võtmeelement. See aitab luua õppimiseks soodsa keskkonna ja loob õpilastele optimaalsed tingimused õpieesmärkide saavutamiseks. Selline lähenemine aitab parandada õppimise efektiivsust ja tagab kasutaja maksimaalse kaasatuse õppeprotsessi.

4.3.2 Interaktiivsus

Interaktiivsus on loodud veebiplatvormi üks põhiomadusi. Töö eesmärk oli luua kasutajate ja keskkonna vahel võimalikult mugav ja tõhus suhtlus, pakkudes kasutajatele laia valikut interaktiivseid elemente.

Saidil kasutusel on laadimise ikoon, mida kuvatakse serverilt vastust oodates. Laadimisikoon asetatakse selliste plokkide peale nagu nupud, tabelid (mis sisaldavad pärast serverilt vastuse saamist andmeid) ja isegi terved lehed. See element annab kasutajatele teavet saidi hetkeseisu kohta ja informeerib neid operatsioonide tulemuste ootamise ajal.

Modaalaknaid kasutatakse kasutajakogemuse parandamiseks ning efektiivsuse ja mis kõige olulisem sisuga töötamise lihtsuse tagamiseks. Modaalaknaid kasutatakse aktiivselt nii klassikalistes ülesannetes (autoriseerimise modaalaken, hoiatusaken) kui ka konkreetsete ülesannete puhul, näiteks ülesande andmebaasi tabelites olevate andmete kuvamiseks või ülesande täitmise oodatava tulemuse kuvamiseks. Kõik need funktsioonid annavad võimaluse kuvada teavet dünaamiliselt ilma kasutaja põhilist töövoogu katkestamata.

Sait pakub rippmenüüplokke, mis annavad kasutajatele võimaluse kuvada dünaamiliselt lisateavet ilma kasutajaliidest üle koormamata. Neid plokke kasutatakse võimalike lahenduste ja vihjete, kirjelduste ja andmebaasi struktuuri kuvamiseks, võimaldades kasutajatel hõlpsasti juurde pääseda vajalikule teabele just sellisel kujul, nagu nad seda vajavad, säilitades samas lihtsuse ja kasutusmugavuse.

Andmete visualiseerimiseks kasutatakse erinevaid interaktiivseid elemente, näiteks sektordiagramme (näiteks statistika lehel) ja jooniseid andmebaasi struktuurist. Need elemendid aitavad kasutajatel esitatud teavet paremini mõista ja teha teadlikumaid otsuseid.

Ikoonide kasutamine on kujunduse oluline aspekt, mis muudab saidi lugemise ja navigeerimise lihtsamaks. Need aitavad visuaalselt eristada liidese elemente, kuvada ülesande täitmise olekut ja hõlbustavad lehel liikumist.

Erilist tähelepanu pööratakse sujuvale suhtlusele saidi elementidega. Kõik avanevad väljad, modaalaknad ja muud interaktiivsed elemendid on realiseeritud sujuvate animatsioonidega, mis muudab kasutajakogemuse meeldivamaks ja mugavamaks.

Erinevate interaktiivsete elementide juurutamine loodud veebiplatvormi peaks aitama parandada kasutajakogemust, muutes selle kaudu õppe- ja praktikaprotsessi efektiivsemaks ja mugavamaks.

4.3.3 Informatiivsus

Informatiivne tekst on loodud veebiplatvormi oluline aspekt. Keskkond püüab kasutajatele pakkuda võimalikult kasulikku teavet, mis aitaks neil keskkonda võimalikult tõhusalt kasutada.

Loodud platvormi üheks omaduseks on administraatori võimalus kasutada erinevates tekstiplokkides esitatavates kirjeldustes HTML-märgendeid (näiteks kogumi kirjeldus, ülesande kirjeldus, vihje kirjeldus, andmebaasi struktuuri kirjeldus jne). See võimaldab näiteks teatud tekstiosasid esile tõsta või lisada linke. See lähenemisviis aitab parandada info loetavust ja võimaldab seda linkide kaudu rikastada.

Administraatorile selle võimaluse andmine pakub talle sisu loomisel rohkem paindlikkust, võimaldades tal seda kasutajate konkreetsetele vajadustele kohandada. See parandab ka õppimist, kuna kasutajatele suunatud teabe kõige olulisema osa saab esile tõsta ja kasutajad saavad järgida linke, et teema kohta lisateavet saada.

Seega on teksti informatiivsus loodud platvormil tagatud mitte ainult kvaliteetse sisuga, vaid ka administraatori poolt HTML-i märgendite kasutamise võimalusega. See lähenemisviis võimaldab administraatoril luua interaktiivsemat ja kaasahaaravamalt sisu, võimaldades tõhusamat kasutajate koolitamist.

4.3.4 SQL kood

SQL-koodi kirjutamise mugavusel on keskkonnas võtmetähtsus. Keskkond püüab pakkuda kasutajatele võimalikult mugavamalt ja tõhusamat viisi SQL lausetega töötamiseks.

Üks platvormi olulisi aspekte on SQL lausete esitamine võimalikult loetaval kujul. Kasutajate enda käivitatud SQL lausete vaatamisel või võimalike vastuste uurimisel vormindatakse kood automaatselt parema loetavuse huvides. See hõlmab vajalike taanete, reavahede ja süntaksi esiletõstmise lisamist, mis muudab SQL lause lugejale hõlpsamini loetavaks ja analüüsivaks.

Kasutajate mugavuse huvides on lausete kirjutamise jaoks kasutusel erinevate funktsioonidega SQL-koodiredaktor. See redaktor toetab süntaksi esiletõstmist, funktsioonide ja märksõnade vihjeid, samuti koodiridade kuvamist ja süntaksivigade

tuvastamist. Tänu nendele funktsioonidele saavad kasutajad SQL-koodiga mugavalt töötada, saades õppe- ja harjutamisprotsessi käigus vajalikku abi ja tuge.

Seega pakub loodud platvorm kasutajale mugavat ja tõhusat viisi SQL-koodiga töötada. SQL lausete vormindamise ja erinevate funktsioonidega redaktori kasutamise võimaluse pakkumine aitab kasutajatel paremini SQL lauseid kirjutada ja analüüsida.

4.3.5 Kasutajaliidese kohandamine lõppkasutaja poolt

Loodud veebiplatvormis on vähesel määral võimalik kasutajaliidese kohandamine. Selle eesmärk on pakkuda kasutajatele mugavust ja individuaalset lähenemist keskkonna kasutamisele.

Üks kasutajaliidese kohandamise viise on ülesande täitmise lehe paigutuse muutmine keskel asuva liuguri abil. See lähenemine võimaldab kasutajatel muuta kahe liideseploki laiust vastavalt oma eelistustele, pakkudes lehe mugavamat ja isikupärasemat kasutamist. Näiteks, kui kasutaja on probleemi kirjeldusega juba tuttav, saab ta seda lehe osa oluliselt vähendada, muutes probleemi lahendamiseks mõeldud osa suuremaks.

Samuti saab kasutaja avada ja sulgeda lehe sektsioone, kus näeb ülesande kirjeldust, tabelite struktuuri illustratsiooni, vihjeid ja võimalikke lahendusi.

Autorite eesmärk ei ole pakkuda liidese kohandamiseks palju võimalusi. Selle põhjuseks on soov säilitada saidi lihtsus ja kasutusmugavus kõigi kasutajate jaoks. Liidese liigse kohandamise pakkumise vältimine aitab vältida teabe üleküllust ja keerukust, mis on eriti oluline haridusplatvormi puhul, kus rõhk peaks olema pigem õppeprotsessil kui liidese seadistustel.

Seega, kuigi liidese kohandamiseks pakutakse mõningaid võimalusi, säilitatakse selles valdkonnas mõõdukus, keskendudes sellele, et platvorm jääks kõigi kasutajate jaoks lihtsaks, mugavaks ja tõhusaks .

4.3.6 Ligipääsetavus

"Veebilehtede ligipääsetavus tähendab, et võimalikud paljud inimesed saavad neid lehti kasutada" [15] [16].

Ligipääsetavuse tagamiseks on aga vaja ka funktsionaalsust erinevates veebilehitsejates testida. Testimise osana testiti esilehte, kuna see on kasutajate sisenemispunkt. Testimine hõlmas modaalakende avanemise kontrollimist, probleemi lahendamise proovimist, statistika vaatamist jne. Testi tulemusena selgus, et süsteem on kasutatav enamikes populaarsetes veebilehitsejates, nagu Chrome 122, Edge 122, Firefox 124, Opera 108, Brave 1.63 ja Safari 5.1.5. See tagab, et kasutajad saavad platvormi vabalt kasutada, olenemata kasutatavast veebilehitsejast, mis omakorda aitab laiendada kasutajaskonda ja tõsta veebiplatvormi ligipääsetavust.

HTML-i analüüs

Algseks veebisaidi HTML5 standarditele vastavuse ja vigade tuvastamise kontrolliks kasutati teenust Html Checker [17]. Kõik leheküljed teisendati veebilehitsejas HTML-formaati, et saaks faile mugavalt teenusesse üles laadida ja tagasisidet saada. Tabelis 1 on toodud iga lehe kontrolli tulemused.

Tabel 1. HTML Checkeriga lehtede kontrollimise tulemus.

Lehekülg	Tulemus
Meist	Olulisi vigu ei leitud.
Ajalugu	Leiti hoiatusi, mis on tingitud pealkirja siltide puudumisest section plokis ja kasutamata aria-label siltidest. Samuti tuvastati vead, mis on seotud atribuudi maxlength valesti kasutamisega arvutüüpi sisendi puhul. Need hoiatused ja vead on tingitud sisseehitatud kalendrimooduli kasutamisest, mis pole töö autorite poolt arendatud.
Ülesande täitmise ajalugu	Leiti viga, mis on seotud id="task_name" sildi duplikaadiga, mida kasutati serverist saadud sisu lisamiseks veebisaidile. Selle asemel tuleks kasutada klasse. Samuti tuvastati vead, mis on seotud väärtuse atribuudi määramisega textarea elemendile, h2-h6 siltide puudumisega section ploki sees ning textarea elemendile keelatud atribuutide autocorrect ja cm-not-content kasutamisega. Need viimased vead on tingitud välise CodeMirror teegi kasutamisest, mis pole saidi autorite tehtud.
Ülesannete loetelu	Olulisi vigu ei leitud.
Ülesande täitmine	Vead on samad, mis tuvastati lehel "Ülesande täitmise ajalugu" samadel põhjustel.
Statistika	Olulisi vigu ei leitud.
Esileht	Ainus viga on seotud sobimatu atribuudi href kasutamisega button sildi sees. See atribuut on kasutusel, et suunata kasutaja ülesannete loetelu lehele.

HTML-i analüüs näitas, et kuigi veebisaidil esineb vigu ja hoiatusi, mis on seotud erinevate elementide valesti kasutatud atribuutidega, ei takista need kasutajal saidi funktsioone kasutamast. Need atribuudid võivad mängida olulist rolli juhul, kui sait tehakse kättesaadavaks laiale avalikkusele ja otsingurobotid seda indekseerivad, et parandada saidi leidmist otsingupäringute kaudu. Kuid kuna loodud tarkvara on mõeldud ainult Tallinna Tehnikaülikooli üliõpilastele, siis ei ole see veebirakenduse aspekt kohustuslik.

Kontrastsuse kontroll

Üldjoontes on veebisaidi värvipalett tasakaalustatud ja kogu tekst on hästi loetav. Saidil on nii musta tausta ja valge tekstiga plokkke kui ka valge tausta ja musta tekstiga plokkke. Kontrastsuse lisakontrolliks Contrast Checkeri abil [18] valiti juhuslikult järgmised plokkid: rippmenüüd, teksti sisestamise väljad ja tekstiplokkid ülesannete täitmise lehel. Kõigi nende plokkide taustavärv on #EBEBEB ja tekstivärv #000000. Nende plokkide kontrastsussuhe on 17,61:1, mis on suurepärane näitaja, kuna väärtused alla 8 võivad tekitada probleeme mõnede tekstisuuruste loetavusega.

4.3.7 Kasutajaliidese kohandamine administraatori poolt

Administraator saab konfiguratsioonifaili kaudu muuta keskkonna pakutavat funktsionaalsust järgmisel viisil (vt ka Joonis 1):

- saab peita kasutajate poolt nende tegevuse ajaloo kustutamiseks mõeldud nupu,
- saab määrata, kui mitu lahenduskatset tuleb teha, et näha vihjeid,
- saab määrata, kui mitu lahenduskatset tuleb teha, et näha võimalikke õigeid vastuseid.

```

config.php > ...
1  <?php
2
3  $config = [
4      "display_errors" => 1,
5      "log_errors" => 1,
6      "version" => "1.1.0.1",
7      "basePath" => __DIR__,
8      "csrfControl" => true,
9      "baseUrlCatalog" => "", // if app is in base server c
10     "template_folder" => "frontend",
11     "defaultLocale" => "est", // language for first-time
12     "config_type" => "LIVE", // DEMO or LIVE - Data for c
13
14
15     // if config_type = LIVE
16     "db_live" => include(__DIR__ . "/db.php"),
17
18     // if config_type = DEMO
19     // "db_demo" same structure, like in db_live
20
21     "system" => [
22         "min_count_to_see_solutions" => -1,
23         "min_count_to_see_hints" => 1,
24         "can_delete_history" => 1 // 1 - can, 0 - cannot
25     ],
26

```

Joonis 1. Config.php fail.

Selliste muudatuste tegemise võimalus lubab õpetajal kasutada seda keskkonda erinevates kasutusstsenaariumites.

4.4 Arendusprotsess ja tulemused

Töö algas sarnaste süsteemidega tutvumisega (vt jaotis 3.1 Olemasolev tarkvara). Seejärel koostati Figma kasutajaliidese prototüüp, mida esitati õppeaine “Andmebaasid I” üliõpilastele. Saadud tagasiside alusel tehti kasutajaliidesesse täiendusi ning seejärel asuti süsteemi realiseerima. Peale esialgse versiooni valmimist ja omapoolset testimist jagati seda juhendajaga, kes pani selle üles õppeserverisse ja testis omalpoolt. Saadud tagasiside alusel tehti programmis mitmeid parandusi ja täiendusi. Samuti tehti tarkvara avalikuks “Andmebaasid I” üliõpilastele ning paluti neil vastata veel ühele küsimustikule. Saadud tagasiside oli aluseks töö tulemuste analüüsimisele ning täiendavate täienduste tegemisele.

4.4.1 Valitud tehnoloogiate kooskõlastamine

Arvestades projekti piiratud ajalisi raame ja nõuet lõppkasutajatele sobiva valmistoote kiireks kasutuselevõtuks, otsustati rakenduse loomisel PHP keele valimise kasuks. See valik on tingitud autorite kogemused PHP-ga töötamisel ja selle laialdasest kasutamises

nende varasemates projektides. Lisaks tagab PHP autorite hinnangul optimaalse tasakaalu arenduskiiruse ja algkoodi kvaliteedi vahel. Vähetahtis polnud ka juhendaja soov olla võimeline rakendust tulevikus väiksemal määral täiendama ja tema sellest tulenev nõusolek kasutada PHP keelt.

Projekti maksimaalse lihtsustamise tagamiseks otsustati mitte kasutada kolmandate osapoolte teeke, sealhulgas neid, mis on saadaval Composer'i kaudu.

Arendamine toimus PHP versioonil 8.0. Andmebaasisüsteemina kasutati juhendaja soovil PostgreSQL 16. Andmebaasisüsteemi valiku põhjusteks on, et see on kasutusel ülikooli andmebaaside õppeainetes, see toetab suures mahus uusimaid SQLi võimalusi ning on samal ajal ka väga populaarne, tasuta ning avatud lähtekoodiga. PostgreSQL andmebaasisüsteemiga suhtlemiseks kasutati järgmisi laiendusi:

- “pgsql” laiendus pakub PostgreSQLiga töötamiseks PHP-s natiivset tuge, tagades tõhusa suhtluse andmebaasiga.
- “pdo_pgsql” laiendus võimaldab PostgreSQLiga suhtlemist läbi PHP andmeobjektide (PDO) [19], tagades andmebaasiga töötamisel mugavuse ja turvalisuse.

4.4.2 Graafilise prototüübi loomine

Tehnoloogiate valikule järgnesid detailsed arutelud veebirakenduse funktsionaalsuse üle ning määrati kindlaks minimaalne hulk vajalikke lehti. Prototüübi loomise rakenduseks valiti Figma, kuna sellel on mahukas funktsionaalsus veebirakenduste detailsete prototüüpide loomiseks ja mugav võimalus prototüübi demonstreerimiseks kolmandatele osapooltele. Töö tulemusena töötati välja järgmised lehed (vt Tabel 2).

Tabel 2. Lehtede kirjeldus.

Lehe nimi	Kirjeldus
Esileht	Siin on üldine teave projekti kohta, samuti sisselogimisfunktsioon ja veebisaidi keele valik.
Ülesannete loendi leht	Võimaldab kasutajatel filtreerida ja otsida huvipakkuvaid ülesandeid ning valida ülesandeid konkreetsest kogumist.
Ülesande täitmise leht	Pakub kasutajatele mugavat liidest ülesannete täitmiseks, sisaldab ülesande kirjeldust, oodatavat tulemust, seotud andmebaasi struktuuri illustreerivat pilti, võimaldab vaadata lähtetabelites olevaid andmeid,

Lehe nimi	Kirjeldus
	esitab vihjeid ülesande lahendamiseks ja võimaldab vaadata võimalikke vastuseid. Kasutaja saab kirjutada ja käivitada lahenduse ning näeb sellele antavat tagasisidet.
Ajaloo leht	Siin saavad kasutajad vaadata kõiki ülesannete täitmise katseid, sorteerides neid täitmise aja järgi ning vajadusel puhastada ajalugu.
Ülesande täitmise ajaloo leht	Pakub teavet viimase ülesande täitmise katse kohta, sealhulgas käivitatud koodi ja süsteemi antud kommentaare, samuti kõigi katsete loendit koos võimalusega tutvuda nende üksikasjadega.
Statistikaleht	Siin saavad kasutajad vaadata üksikasjalikku statistikat ülesannete täitmise kohta süsteemis, samuti täidetud ülesannete protsenti erinevate parameetrite järgi nagu näiteks päringu tüüp, raskusaste ja kogumid.
Programmileht	Sisaldab täiendavat teavet tarkvara ja selle loojate kohta.

4.4.3 Tagasiside kogumine prototüübi kohta

Prototüübi kohta tagasiside kogumiseks töötati välja küsimustik, kasutades Microsoft Forms keskkonda, et hinnata kasutajate rahulolu tulevase tarkvaraga. Küsimustik hõlmas selliseid aspekte nagu visuaalse kujunduse, kasutajaliidese selguse, värviskeemi ja elementide paigutuse hindamine, samuti tekstilise sisu ja muude veebirakenduse kasutatavuse aspektide hindamine.

Saadud tagasiside põhjal analüüsiti ja määrati kindlaks peamised parandamist vajavad aspektid. Vajaduse korral tehti prototüübis kohandusi vastavalt tuvastatud kasutajate vajadustele ja eelistustele. Prototüübi kohta tehtud küsitluse ja selle vastuste analüüsi kohta saab lugeda jaotisest 5.2 Kasutajate poolne kasutajaliidese prototüübi testimine.

4.4.4 Eesrakenduse arendus

Selles etapis loodi staatilise sisuga lehekülgede puhtad HTML-, CSS- ja JavaScript-failid. Peamise stiiliraamatukoguna kasutati Bootstrap v5.3.2 ja JavaScripti jaoks jQuery v3.7.1. Selle etapi tulemusel loodi projekti koopia Figma'st, võttes arvesse dünaamilisi elemente, nagu modaalsed aknad, ripploendid, kerimisblokid jne.

4.4.5 Välised teegid ja komponendid

Veebiplatvormi arendamise käigus kasutati funktsionaalsuse ja hea kasutuskogemuse pakkumiseks aktiivselt erinevaid väliseid teeke ja komponente. Peamised komponendid,

mis mängivad SQL-koodiga töötamisel võtmerolli, on teegid Codemirror ja Sql-formatter.

Codemirror on funktsioonirikas tekstiredaktor, mis on kirjutatud JavaScriptis ja on mõeldud veebirakendustesse manustamiseks [20]. See pakub süntaksi esiletõstmist, automaatset lõpetamist, süntaksivigade kontrollimist ja muid funktsioone, muutes selle ideaalseks tööriistaks SQL-koodiga töötamiseks. Töös kasutati Codemirrorit saidi erinevatel lehtedel, sealhulgas ülesannete lahenduskatsete ajaloo lehtedel ja ülesannete lahendamise lehel, et võimaldada kasutajatel hõlpsasti SQL-lauseid vaadata, luua ja redigeerida.

Sql-formatter on JavaScripti teek SQL-koodi vormindamiseks loetavamale kujule [21]. See lisab taande ja reavahed, mis muudab SQL-koodi arusaadavamaks ning hõlpsamini loetavaks ja analüüsitavaks. Seda kasutatakse koos Codemirroriga, et vormindada automaatselt SQL-koodi ülesannete ajaloo lehtedel ja probleemide lahenduse lehel.

Flatpickr js-i kalender on kerge ja paindlik teek veebirakendustes kuupäevade ja kellaegadega töötamiseks [22]. See pakub kasutajasõbralikku kuupäeva ja kellaaja valimise liidest ning toetab ka erinevaid võimalusi kalendri välimuse ja käitumise kohandamiseks. Loodud vahendis kasutati seda teeki kõigi ülesannete lahenduskatsete ajaloo lehel, et kasutaja saaks filtreerida katseid algus- ja lõppaja järgi. See võimaldas kasutajatel hõlpsalt leida ja analüüsida oma katseid ülesande täitmiseks teatud ajavahemike jooksul.

Lightgallery js on mugav ja kerge teek pildigaleriide loomiseks, mis toetab erinevaid efekte ja kohandamisvalikuid [23]. See pakub lihtsat liidest piltide vaatamiseks mugavas vormingus, sealhulgas võimalust pilte sisse ja välja suurendada ning alla laadida. Loodud vahendis kasutati seda teeki ülesande lahendamise lehel andmebaasi struktuuri joonise näitamiseks. Andmebaasi struktuuri joonisel klõpsates avaneb suurem täisekraani aken, kus kasutajad saavad andmebaasi struktuuriga tutvuda ja seda täpsemalt analüüsida. See andis mugavama ja visuaalsema viisi platvormi oluliste elementidega töötamiseks.

4.4.6 SQL koodi kontrollimine

Kasutaja ülesande SQL-koodi kontrollimine koosneb mitmest etapist.

Enne kui kasutaja koodi tema isiklikus andmebaasis käivitatakse, tuleb koodiga läbi viia mitmeid kontrole, et see vastaks ülesandele ja süsteemi globaalsetele reeglitele. Siin on tingimused, mille korral koodi ei peeta kehtivaks, st seda ei käivitata kasutaja andmebaasis:

- Kui lause ei vasta ülesande tüübile (ei sisalda nõutud mustrit või sisaldab keelatud mustrit).
- Kui lause vastab vähemalt ühele üldisele aktiivsele mittelubatud mustrile.
- Kui lause ei vasta kõigile aktiivsetele ülesande-spetsiifilistele nõutud mustritele.
- Kui lause vastab vähemalt ühele aktiivsele ülesande-spetsiifilisele mittelubatud mustrile.
- Kui INSERT, UPDATE, DELETE või MERGE lause on vale tabeli põhjal.
- Kui SELECT lauses, mille puhul sorteerimine on nõutud (`is_sorting_required=TRUE`), ei ole sorteerimist.

Nende tingimuste kontrollimiseks on andmebaasis funktsioonid, mis ühtlasi registreerivad leitud vead.

Kui esimesel etapil vigu ei leita, siis kood käivitatakse ja koodi õigsuse kontrollimine toimub saadud vastuse põhjal.

Esmalt loetakse andmebaasist ülesande õige lahendus, et võrrelda õige lahenduse ja kasutaja lahenduse täitmise tulemusi. Pärast vajalike andmete salvestamist muutujatesse toimub täitjarolli uuesti ühendamine kasutaja andmebaasiga. Kasutaja andmebaasi nimeks on `t[user_identificator]`. Kui andmebaas oli juba süsteemis loodud, siis see kustutatakse ja luuakse uuesti konkreetse ülesandega seotud andmebaasi malli alusel. Andmebaasi uuesti loomine on vajalik, sest erinevad ülesanded on erineva struktuuriga andmebaaside põhjal ning kasutaja saab neid läbisegi lahendada. Samuti võis kasutaja eelnev ülesande lahendamise katse jätta andmebaasi seisu, kus see ei ole enam samasugune nagu algne andmebaas (nt mingid read või tabelid on kustutatud). Pärast seda jätkub kontroll sõltuvalt lause tüübist kolmel võimalikul viisil.

SELECT lause

SELECT lause loetakse õigeks kui see läbib kõik algsed kontrollid ja selle käivitamise tulemuseks saadud tabel on samasugune (sh ridade ja veergude järjekord; veergude nimed on ühesugused või sarnased) kui õigeks loetud lahenduse käivitamise tulemus.

Selle lausetüübi jaoks luuakse esmalt CREATE TABLE .. AS lausega tabelid A ja B, mis põhinevad vastavalt ülesande õige lahenduse ja kasutaja SQL-koodi tulemustel. Pärast seda toimub kaks kontrollietappi:

Esimene etapp keskendub kahe päringu tulemusena saadud tabelite veergude sarnasuse tuvastamisele. Kontrollide läbiviimiseks käivitatakse SQL lauseid.

Järgnevate kontrollide SQL laused käivitatakse süsteemikataloogi põhjal, et võrrelda tabelite A ja B struktuuri.

- Kas tabelite A ja B veergude arv on ühesugune?
- Kas tabelites A ja B on sama nimega ja sama tüüpi veerud samas järjekorras?
- Kas tabelites A ja B on sama tüüpi veerud samas järjekorras?
 - Kas tabelites A ja B on veergude nimed sarnased?
- Kas tabelites A ja B on sama nimega ja tüüpi veerud (järjekord pole oluline)?

Teises etapis kontrollitakse päringu tulemuse tabelite sisu õigsust. See etapp sisaldab järgmisi kontrole:

- Kas tabelid A ja B sisaldavad samu ridu (järjekord pole oluline)?
- Kas tabelite A ja B ridade järjekord on ühesugune?

INSERT, UPDATE, DELETE või MERGE laused

INSERT, UPDATE, DELETE ja MERGE lause loetakse õigeks kui see läbib kõik algsed kontrollid ja selle käivitamise tulemusena on muudetud andmetega tabelis samasugused andmed kui õigeks loetud lahenduse käivitamise tulemusena.

INSERT, UPDATE, DELETE ja MERGE lausete puhul eeldatakse, et muudetakse ühte tabelit (see tabel on ülesande juures määratud) – tähistame seda tähega T. Luuakse tabelid A ja B, kus A esitab tabelis T olevaid andmeid andmebaasist loetud õige lause täitmise järel ja B esitab tabelis T olevaid andmeid kasutaja käivitatud lause täitmise järel (vt Joonis 2). Nende loomiseks kasutatakse võimalust luua ajutine tabel mille nimi on

samasugune nagu tabelil T ning teha andmemuudatus loodud ajutise koopia, mitte originaali põhjal.

```
SET search_path=pg_temp, public;
DROP TABLE IF EXISTS A;
DROP TABLE IF EXISTS B;
CREATE TEMPORARY TABLE [task_related_table] AS SELECT * FROM [task_related_table];
[possible_solution_sql];
CREATE TABLE A AS SELECT * FROM [task_related_table];
DROP TABLE [task_related_table];
CREATE TEMPORARY TABLE [task_related_table] AS SELECT * FROM [task_related_table];
[user_sql];
CREATE TABLE B AS SELECT * FROM [task_related_table];
DROP TABLE [task_related_table];
```

Joonis 2. Tabelite B ja A loomine vastavalt kasutaja otsusele ja probleemi võimalikule lahendusele.

Pärast tabelite A ja B loomist tehakse lõplik päring, et kontrollida, kas tabelid A ja B kattuvad (seal on samad read). Kui nad kattuvad, tagastab päring TRUE, mis näitab kasutaja koodi õigsust. Vastasel juhul tagastatakse FALSE. Selles etapis võib tekkida kahte tüüpi vigu: Lause tulemus on ebatäpne ja serveriviga, mis tähendab, et koodi ei õnnestunud üldse käivitada.

CREATE TABLE, ALTER TABLE või DROP TABLE laused

CREATE TABLE, ALTER TABLE ja DROP TABLE lause loetakse õigeks kui see läbib kõik algsed kontrollid ja ükski ülesandega seotud test ei ebaõnnestu.

Seda tüüpi lausete ülesannete jaoks on andmebaasis ette nähtud testid, mille sisuks on lausete täitmine muudetud struktuuriga andmebaasi põhjal. Need toimuvad järgmiselt: kõigepealt käivitatakse kasutaja kood, seejärel käivitatakse kontrollkood. Sõltuvalt sellest, milline peaks olema testi tulemus (kas edukas täitmine või vea tekkimine), loetakse test kas läbituks või mitte. Näiteks kui testi SQL lause täitmine peaks põhjustama vea ilmnemise, kuid tegelikult viga ei ilmne, siis loetakse test ebaõnnestunuks. Sama kehtib vastupidises olukorras, kui test eeldab edukat tulemust. Testide käigus ebaõnnestunud täitmine salvestatakse ja juhul, kui vähemalt üks test on ebaõnnestunud, saab kasutaja loetelu nendest testidest ning kui testil on kirjeldus, siis näeb kasutaja ka seda kirjeldust.

4.4.7 Tagarakenduse arendus

Paralleelselt eesarenduse loomisega algas veebirakenduse serveripoolse loogika loomine. Kolmandate osapoolte teekide mittekasutamise tõttu realiseeriti kõik marsruudid, kasutaja sessiooni haldamise loogika, mitme keele tugi ja muud standardsed veebirakenduse funktsioonid iseseisvalt.

Võttes arvesse eeldatavat aktiivset suhtlemist serveri ja kliendiosa vahel, valiti andmeedastuse tarkvaraarhitektuuri stiiliks REST API, kasutades MVC (Model-View-Controller) disainimustrit. REST API mudelis kasutatakse selliseid meetodeid nagu GET (andmete hankimiseks serverist), POST (uue ressursi loomiseks), PUT (olemasoleva ressursi uuendamiseks) ja DELETE (ressursi kustutamiseks serverist).

Selle tulemusena struktureeriti projekt 25-ks lõpp-punktiks, seitsmeks kontrolleriiks ja seitsmeks mudeliks. Täpsem kirjeldus, millised lõpp-punktid on loodud ja mis mille eest vastutavad, on kirjas Tabel 3, Lisa 2.

4.4.8 Ees- ja taga rakenduse ühendamine

Pärast tagarakenduse lõpp-punktide loomise lõpetamist alustati nende integreerimist eesrakendusega. Selle protsessi käigus realiseeriti serverilt saabuvate eri-tüüpi vastuste töötlemine: ootamatu vea korral, serveril töödeldud vea korral (vastava veateksti kuvamine) ja eduka taotluse korral (sõnumi või saadud andmete kuvamine). Samuti lisati integratsiooniprotsessi käigus erinevaid interaktiivseid elemente, näiteks allalaadimise ikoonid, et parandada kasutajakogemust.

4.4.9 Testimine

Projekti testimiseks kasutati erinevaid lähenemisviise. Kõikide lõpp-punktide testimiseks loodi Postmanis API testid. Kokku loodi neid 49: 25 testi kõigi lõpp-punktide kontrollimiseks ja kolm lisatesti iga lause tüübi jaoks (SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE ja DROP TABLE). Lisatestid on järgnevad:

- Tehakse päring oodatud tulemuse saamiseks, kontrollimaks, et kasutaja saab serverist õiges vormis oodatud tulemuse. SELECT, INSERT, UPDATE, DELETE, MERGE puhul tagastatakse tulemuseks tabeli õigete andmetega.

- Tehakse päring, mis sisaldab ülesande lahendamiseks õiget lauset, kontrollimaks, et kasutaja saab serverist õiges vormis oma SQL koodi täitmise tagasiside (et tema lahendus on õige).
- Tehakse päring, mis sisaldab vale lauset ülesande lahendamiseks, et kontrollida, et kasutaja saab serverist õiges vormis oma SQL koodi täitmise tagasiside (et tema lahendus on vale ja sellega seotud vead/kontrollid/sõnumid) ja tulemuse (kui tema SQL koodi oleks võimalik käivitada, siis andmekäitluskeele lause tulemusena saab kasutaja tabeli).

Pärast kõigi komponentide – nii tagarakenduse kui ka eesrakenduse – valmimist pandi tarkvara privaatserverisse, et kõik meeskonnaliikmed saaksid mugavalt käsitsi funktsionaalset testimist teha. Iga testimise etapp hõlmas saidi täielikku läbikäimist, et tuvastada nõrkused ja vead. Leitud probleemid lahendati ja vajadusel lisati uued ülesanded GitLab versioonihaldussüsteemi edasiseks arendamiseks. Funktsionaalsuse käsitsi testimise käigus [24] suhtlesid meeskonna liikmed aktiivselt omavahel, arutasid avastatud probleeme ja otsustasid järgmiseid samme. Omavaheline tihe suhtlus võimaldas tarkvara kvaliteeti põhjalikumalt hinnata ning võtta arvesse kõiki märkusi ja soovitusi. Pärast testimise lõpetamist analüüsiti tulemusi, parandati tuvastatud probleemid ja pöörduiti vajaduse korral tagasi arendusetappi. See tsükkel tagas tarkvara pideva täiustamise ja hea kvaliteeditaseme saavutamise.

4.4.10 Väljalase ja tagasiside kasutajatelt

Pärast projekti kõigi peamiste etappide lõpuleviimist migreeriti autorite arvutites loodud PHP-projekt ülikooli serverisse. Ülekandmine toimus programmi koodiga zip-arhiivi ülekandmise teel. Tarkvara edukaks installeerimiseks oli saidi administraatoril vaja vaid luua andmebaas ja teha vajalikud kohandused projekti konfiguratsioonifailis (*config.php*). Selles failis saab teha muudatusi, mis on seotud projekti asukohaga serveris, töörežiimiga („demo” või „live”) ja saidi põhikeelega. Samuti kirjeldati failis ühendus andmebaasiga, määrates tee serveris oleva failini, mis sisaldab seda informatsiooni. Seda tehti selleks, et piirata juurdepääsu tundlikele saidi andmetele nii palju kui võimalik.

Pärast projekti seadistamise lõpetamist serveris ja süsteemi uuesti testimist ülikooli serveris, loodi teine tagasiside küsitlus, et saada tagasisidet üliõpilastelt, kes olid süsteemi testinud. Saadud tagasiside alusel tehti rakenduses veel hulk parandusi. Rakenduse kohta

tehtud küsitluse ja selle vastuste analüüsi kohta saab lugeda jaotisest 5.4 Kasutajate poolne rakenduse testimine.

5 Analüüs ja järeldused

Turul on mitmeid platvorme, mis pakuvad SQL õppimise ja harjutamise võimalusi. Näiteks, zyBooks [25] ja SQLZoo [26] keskenduvad interaktiivsele õppekogemusele, samal ajal kui HackerRank[6], SQLPad [9] ja LeetCode [7] ning teised pakuvad keskkondi [27] [28] [29] [30], kus kasutajad saavad SQL lauseid reaalsajas kirjutada ja käivitada. Vaatamata nende platvormide laiale kättesaadavusele ja kasulikkusele, on neis mitmeid piiranguid ja väljakutseid, millest lähtuvalt on ruumi uue, paremini integreeritud ja kasutajasõbraliku veebikeskkonna loomiseks.

5.1 Võrdlus olemasoleva tarkvaraga

Lõputöö tegemine algas olemasolevate SQL õppimise ja harjutamise veebikeskkondade uurimisega, et analüüsida nende eeliseid ja puudusi (vt jaotis 3.1 Olemasolev tarkvara). Selle analüüsi eesmärk oli välja selgitada, millised funktsioonid on tõhusad ja mis aspektid vajavad täiustamist. Uurimistöö tulemused aitasid kujundada uue keskkonna, mis vastab paremini kasutajate vajadustele, pakkudes tõhusamat ja interaktiivsemat õppimiskogemust. Koondtabel loodud vahendi ja jaotises 3.1 esitatud vahendite mõningate põhiomaduste võrdlusega esitatakse lisas 11.

5.1.1 Kasutajaliidese eelised võrreldes olemasoleva tarkvaraga

Kui vaatleme kasutajaliidese eeliseid võrreldes olemasoleva tarkvaraga, siis on mõned olulised tegurid, mis tõstavad esile loodu vahendi ainulaadsust ja kasutajasõbralikkust.

Platvorm toetab erinevaid kasutajaliidese ja ülesannete keeli. Tänapäeva üleilmastuvas maailmas on mitmekeelsus ja kohandatavus olulised komponendid, mis teevad tehnoloogilised lahendused kättesaadavamaks ja kasulikumaks suuremale hulgale eri kultuuritaustaga kasutajatele. Erinevalt paljudest teistest tarkvaradest, mis võivad olla piiratud kindla keelega või pakkuda vaid piiratud lokaliseerimisvõimalusi, rõhutab antud platvorm keeleliste barjääride ületamist ja kasutajate kaasamist. See lähenemine mitte ainult ei laienda platvormi võimalikku kasutajaskonda, vaid aitab kaasa ka üldisele kasutajakogemuse parandamisele, muutes selle eriti atraktiivseks nii rahvusvahelistele kui kohalikele kasutajatele.

Erinevad tarkvarad piirduvad sageli lihtsate veateadetega, mis ei pruugi pakkuda piisavat selgust ega juhiseid probleemi lahendamiseks. Vastandina sellele pakub käesolev süsteem vigade põhjalikku selgitust ja võimalust näha õigeid lahendusi, mis toetab sügavamat mõistmist ja parandab õppekvaliteeti.

Veel üheks oluliseks eeliseks on see, kuidas kasutajaliides andmeid kuvab. Selgelt esile toodud tabelid ja võimalus näha lausete tulemusi reaalajas on andmebaasi struktuuri ja käitumise mõistmisel suureks abiks. Lisaks aitavad kasutajaliideses pakutavad vihjed mõista ja lahendada keerukamaid ülesandeid, muutes need arusaadavamaks suuremale hulgale kasutajatele. Selline läbimõeldud ja kasutajasõbralik disain tagab, et kasutajad ei tunne end vahendit kasutades jõuetuna, vaid on varustatud infoga, mida nad vajavad edukaks ja tõhusaks tööks.

5.1.2 Funktsionaalsuse eelised võrreldes olemasoleva tarkvaraga

SQL harjutuskeskkond pakub funktsionaalsuse osas mitmeid olulisi eeliseid turul olemasolevate lahenduste ees. Rakendus sisaldab intuiitivset võimalust vaadata ülesannete kogumeid, mis lihtsustab kasutajatele huvipakkuvate teemade ja ülesannete otsimist. Selle süsteemi efektiivsus peitub võimaluses leida kogumite abil kiiresti ja hõlpsalt kasutajatele sobivad ülesanded. See toetab aktiivset õppimist ja kaasatust. Lisatud filtreerimissüsteem võimaldab kasutajatel valida ülesandeid nende keerukusastme ja lausetüübi järgi. See funktsionaalsus mitte ainult ei suurenda kasutusmugavust, vaid võimaldab ka iga kasutaja individuaalsete vajaduste ja eelistustega arvestavat õppimiskogemust.

Vahendisse saab kerge vaevaga lisada uusi ülesandeid (vastava kasutajaliidese loomine ei kuulunud küll käesoleva töö skoopi, kuid selline liides on administraatori poolt loodud), andes administraatorile/õpetajale võimaluse vahendit oma vajadustele vastavalt kohandada. Vahendi pakutav tasuta SQL-i harjutamise võimalus on eriti märkimisväärne, arvestades, et enamik konkureerivaid veebirakendusi piirdub peamiselt andmete otsimise ülesannetega ja SELECT-lausete kasutamise võimaldamisega. Erinevalt tavapärasest lähenemisest, kus kasutajad on piiratud andmete otsimisega, võimaldab loodud vahend õpilastel, kes osalevad andmebaaside kursustel Tallinna Tehnikaülikoolis, laiendada oma teadmisi ja oskuseid palju sügavamal tasemel. See unikaalne funktsioon hõlmab SQL-i laia spektrit, alates andmete sisestamisest (INSERT) ja uuendamisest (UPDATE) kuni baastabelite loomise (CREATE TABLE), struktuuri või käitumise muutmise (ALTER

TABLE) ja kustutamiseni (DROP TABLE). Sellise mitmekülgse praktika kaudu saavad õpilased omandada tõeliselt praktilisi oskusi, mis on hädavajalikud professionaalses töökeskkonnas toimetulekuks.

Rakendus kogub lahendamise andmeid, mis loob õpetajatele võimaluse jälgida ja analüüsida õpilaste ülesannete lahendamise statistikat. See statistika võib hõlmata informatsiooni, mis puudutab levinumaid vigu, võimaldades õpetajatel oma õppemethodikat vastavalt vajadusele kohandada. See omakorda suurendaks õpetamise efektiivsust ja aitaks kaasa õpilaste paremale mõistmisele ja oskuste arengule. Kuid ka õppijatele on loodud statistikasüsteem, mis võimaldab neil jälgida oma edusamme, analüüsides tehtud katsete arvu, edukate ja ebaedukate katsete suhet ning ülesannete lõpuleviimise määra. See süsteem aitab kasutajatel mõista oma arengut ja tuvastada valdkondi, kus on vaja täiendavat harjutamist või õppimist. Kasutajatel on võimalus oma lahenduste ajalugu kustutada. Süsteemi siseselt tähendab see lahenduste juurest kustutatakse lahendaja identifikaator. Nii saaks õpetaja näiteks endiselt vaadata infot sagedaste vigade kohta.

Tagasiside funktsioon, mis võimaldab kasutajatel hõlpsalt teha märkuseid ülesannete kohta ja saada õpetajatelt vastuseid, loob täiendava suhtluskanali õpetajate ja õpilaste vahel. See suhtlus on oluline, tagades, et iga õpilase vajadused ja küsimused saavad kiiresti ja tõhusalt lahendatud, pakkudes nii personaalsemat toetust õppeprotsessile.

5.1.3 Funktsionaalsuse puudused võrreldes olemasoleva tarkvaraga

Vaatamata sellele, et loodud veebipõhine rakendus pakub kasutajatele olulisi praktilisi õppevõimalusi, tuleb tunnistada, et sellel on ka teatud funktsionaalsed piirangud, mida ei saa eirata. Üheks oluliseks puuduseks on rakenduse piiratud toetus erinevatele andmebaasisüsteemidele. Praeguses versioonis puudub ühilduvus selliste laialdaselt tunnustatud andmebaasisüsteemidega nagu näiteks MySQL, Microsoft SQL Server ja Oracle. See piirang võib vähendada rakenduse kasutatavust mitmesugustes akadeemilistes ja professionaalsetes kontekstides, kus nimetatud süsteemid on eelistatud.

Lisaks on vahendi suunatus praktilisele õppele toonud kaasa teoreetiliste materjalide rakenduse sees kasutamise võimaluse puudumise. Ainus võimalus on lisada ülesannete või vihjete tekstidesse linke välistele veebilehtedele. Selline lähenemisviis võib piirata kasutajate võimet saavutada sügavam arusaam SQL keelest ja andmebaaside

optimeerimisest, mis on kriitilised oskused keerukate andmebaasi probleemide lahendamisel.

Rakendus ei paku kasutajatele edetabelit ega reitingusüsteemi, mille olemasolu võib suurendada kasutajate kaasatust ja pakkuda lisaväärtust motivatsiooni tõstmise kaudu.

Samuti ei paku rakendus võimalust SQL-koodi failidest otse laadida, mis on funktsioon, mida pakuvad mõned konkurendid ja mis võib parandada kasutusmugavust ja töövoogude efektiivsust neile kasutajatele, kes seda võimalust vajavad. Arvestades, et loodud tarkvara on mõeldud korraga ühe lause käivitamiseks, ei ole see siiski kriitiline puudus.

On oluline märkida, et kuigi nende funktsioonide puudumine ei takista rakenduse kasutamist, siis võib kaaluda nende arendamist tulevikus.

5.2 Kasutajate poolne kasutajaliidese prototüübi testimine

Enne prototüübi arendamist otsustati koostada küsimustik, et koguda süsteemset tagasisidet arendamisel oleva SQL-i harjutamise keskkonna kohta. Küsimustiku koostamise keskkonnaks valiti Microsoft Forms.

Küsitlus koostati eesmärgiga saada üksikasjalikku tagasisidet prototüübi erinevate aspektide kohta, alates kasutajate esmastest muljetest kuni konkreetsete lehtede funktsionaalsuse ja disainielementideni. Selle struktureeritud tagasiside abil sooviti saada selge ülevaade, kuidas planeeritud rakendus vastab kasutajate vajadustele ja kuidas seda edasi arendada. Küsitlus toimus kevadsemestri esimeses pooles, märtsi algul, ajal kui üliõpilased alustasid “Andmebaasid I” praktikumides SQLi harjutamist.

5.2.1 Küsitluse korraldus ja küsimustik

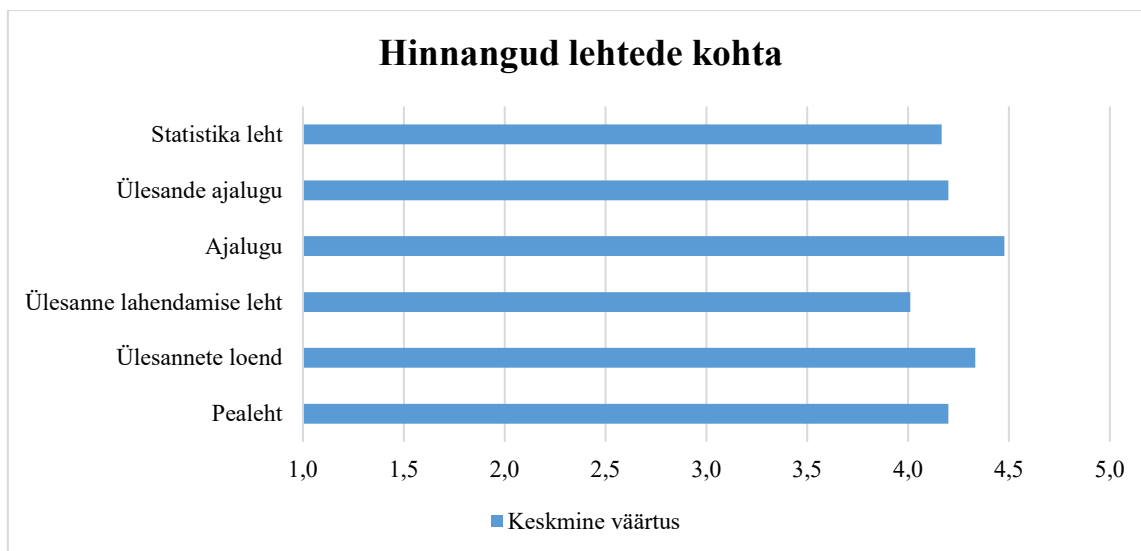
Infot küsitluse kohta jagati õppeaine “Andmebaasid I” kodulehel, eesmärgiga koguda tagasisidet arendamisel oleva SQL-i harjutamise keskkonna kohta. See oli mõeldud süsteemi võimalikele tulevastele kasutajatele, kes soovivad aidata kaasa uue õppetöökeskonna loomisele, pakkudes oma panust prototüübi testimisse ja parandamisse. Alguses paluti kasutajatel tutvuda prototüübiga Figma keskkonnas (vt Lisa 7) [31], kus nad said üle vaadata erinevad leheküljed, katsetada funktsioone ja erinevaid kasutajaliidese elemente.

Küsitluse esimeses osas paluti vastajatel hinnata oma SQL-oskuseid, et mõista paremini nende oskuste taset. Järgnevalt paluti osalejatel anda hinnangud mitmetele rakenduse lehtedele, sealhulgas pealehele, ülesannete loendile, ülesande lahendamise lehele, ajaloo lehele, ülesande ajaloo lehele ning statistika lehele. Hindamiskriteeriumid sisaldasid disaini, elementide interaktiivsust, intuitiivset navigeerimist, lehe struktuuri, esitatud info selgust, visuaalset atraktiivsust, kasutamise lihtsust, pealkirjade ja põhielementide atraktiivsust ning esitatud andmete selgust. Samuti küsiti avatud küsimustega ettepanekuid lehtede parandamiseks.

Lõpuks uuriti kasutajatelt, kas nad kasutaksid platvormi tulevikus, näiteks „Andmebaasi I“ kursuse raames ning kas nad soovitaksid seda oma kolleegidele või sõpradele. See struktuurne ja mitmetasemeline lähenemine võimaldas koguda kasulikku tagasisidet, et luua veelgi kasutajasõbralikum, intuitiivsem ja parem õppekeskkond.

5.2.2 Tulemused

Küsitlusele vastas kokku 10 vastajat. Küsitluse tulemused näitavad, et enamik kasutajaid, kes osalesid SQL-i harjutuskeskkonna prototüübi testimises, hindasid oma SQL-oskuseid madalaks, peamiselt vahemikus 1 kuni 2 (viie palli skaalal). See madal hinnang peegeldab asjaolu, et küsitlus toimus kursuse alguses, mil õpilased alles alustasid oma õpinguid ja olid eriti huvitatud uue õppekeskkonna loomisest. Hoolimata algaja staatusest, hindasid kasutajad platvormi erinevaid aspekte kõrgelt. Disain, interaktiivsus ja visuaalne atraktiivsus said keskmised hinned vastavalt 3,9, 4,2 ja 4,1 (maksimaalne hinnang on 5), mis näitab, et keskkond on visuaalselt köitev ja kasutajasõbralik. Samuti hinnati kõrgelt navigeerimise intuitiivsust ja lehtede struktuuri, keskmiste hinnetega 4,3 ja 4,2, rõhutades kasutusmugavust ja selget paigutust. Teabe selgus sai keskmise hinnangu 4,1. See näitab, et esitatud teave oli arusaadav ja ligipääsetav. Prototüübi uuringu tulemustel põhinevaid tulpdiagramme erinevate lehtede kohta saab vaadata lisast 8.



Joonis 3. Hinnangud lehtede kohta.

5.2.3 Tulemuste analüüs ja nende alusel tehtud muudatused

Vastused näitasid, et enamik katsetajaid olid platvormi tulevikus kasutamise suhtes positiivselt meelestatud, eriti seoses „Andmebaasi I“ kursusega. Kasutajad arvasid, et vahend aitaks neil oma SQL oskuseid parandada, mis on kursuse kontekstis väga oluline. Kasutajatelt saadud tagasiside oli rikkalik ja mitmekesine, kajastades nii platvormi tugevusi kui ka arenguvajadusi. Järgnevalt on välja toodud mõned olulisemad punktid.

Üks kasutaja kiitis vahendit, öeldes: „Vajalik platvorm, hea idee, mulle meeldib funktsioonide hulga ja lihtsuse kombinatsioon,“ mis rõhutab selle vahendi praktilisust ja innovaativsust. Teine kasutaja tunnustas ideed, märkides: „Idee on väga hea ja keskkond kindlasti väga vajalik,“ toonitades vahendi tähtsust SQL-oskuste arendamisel. Samuti tõi kasutaja välja, et „teksti võiks pealehel veidi vähem olla,“ mille peale muudeti pealehe teksti ning programmi kirjeldust ka selgemaks ja konkreetsemaks. Ühe kasutaja kommentaar „Värvivalik teistsuguseks, praegu oli silmadele kuidagi ebameeldiv ja veidi igav ka. Fonti soovitaksin samuti muuta,“ viis ka visuaalsete elementide, nagu fontide ja värvivaliku, uuendamiseni, et parandada üldist kasutajakogemust.

Lisaks andsid kasutajad olulist tagasisidet statistika lehe kohta, märkides, et ajamõõtmine tundus ebaoluline ja soovitati selle asemel kuvada iga kogumi kohta tehtud ja tegemata ülesannete arvu. Selle põhjal eemaldati ajamõõtmine ja lisati soovitud funktsioone. Veel sooviti näha statistikat sektordiagrammi kujul, millele vastavalt tehti muudatusi.

Tänu nendele kasulikele ettepanekutele ja kommentaaridele said töö autorid luua parema ja kasutajasõbralikuma veebirakendus, mis vastab paremini õppijate vajadustele ja eelistustele.

5.3 Kasutajate poolne rakenduse testimine

Valminud rakenduse testimiseks koostati küsimustik kasutades selleks sama Microsoft Forms platvormi, mida kasutati ka prototüübi kohta tagasiside kogumisel.

Küsitlus koostati eesmärgiga saada väärtuslikku tagasisidet veebirakenduse mitmesuguste aspektide kohta, sealhulgas selle vastavuse kohta kasutajate vajadustele ja selle võimaliku edasiarendamise kohta. Küsitluse käigus paluti osalejatel hinnata rakenduse kasutajaliidest, funktsionaalsust ja üldist kasutuskogemust, mis võimaldaks tuvastada kriitilisi aspekte, mida on vaja kas kohe täiustada või kaugemas tulevikus edasi arendada.

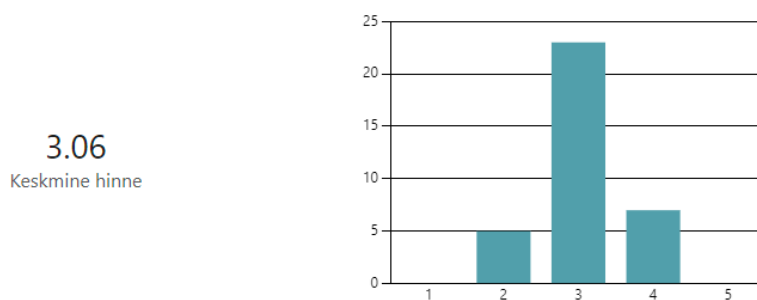
5.3.1 Küsitluse korraldus ja küsimustik

Autorid ja juhendaja levitasid küsitluse kohta infot Tallinna Tehnikaülikooli üliõpilaste seas. Juhendaja pakkus „Andmebaasid I“ õppeaines võimalust koguda keskkonnas kõigi spetsiaalselt selleks koostatud kogumisse kuuluvate ülesannete lahendamise ja seejärel küsitlusele vastamise tulemusel eksamile lisapunkte. Küsitlus toimus aprilli lõpul ja mai alguses – ajal kui õppeaines „Andmebaasid I“ oli SQLi harjutamine juba lõppenud ja paljud üliõpilased olid juba saanud kursuse SQL komponendi eest hinde. Küsitluse esimeses osas paluti vastajatel hinnata oma SQL-oskuseid (viie palli skaalal), et paremini mõista nende oskuste taset. Seejärel paluti osalejatel anda hinnangud mitmetele rakenduse lehtedele, nagu pealeht, ülesannete loend, ülesande lahendamise leht, ajaloo leht ja statistika leht. Iga lehe kohta paluti anda hinnang viie palli skaalal. Küsitluse lõpus uuriti kasutajatelt, kas nad kasutaksid platvormi tulevikus, näiteks "Andmebaasi I" kursuse raames, ja kas nad soovitaksid seda ka oma kolleegidele või sõpradele. Lisaks küsiti, kas on mingeid vajalikke funktsionaalsuseid, mis nende hinnangul puuduvad ning anti võimalus esitada vabas vormis kommentaare või ettepanekuid platvormi täiustamiseks.

5.3.2 Tulemused

Küsitluse perioodil käivitasid keskkonnas 37 erinevat kasutajat (töö autoreid ja juhendajat ei arvestata) lauseid kokku 2194 korda. Käivitamisi, mille korral loeti vastus õigeid, oli kokku 255 ja õigete vastusteni jõudmiseks kulutasid kõik vastajad kokku 24 tundi. Küsitlusele vastas kokku 35 inimest. Vastajate keskmine hinnang oma SQL-oskustele oli 3 (vt joonis 4).

1. Hinnake oma SQL-oskusi 1 kuni 5?

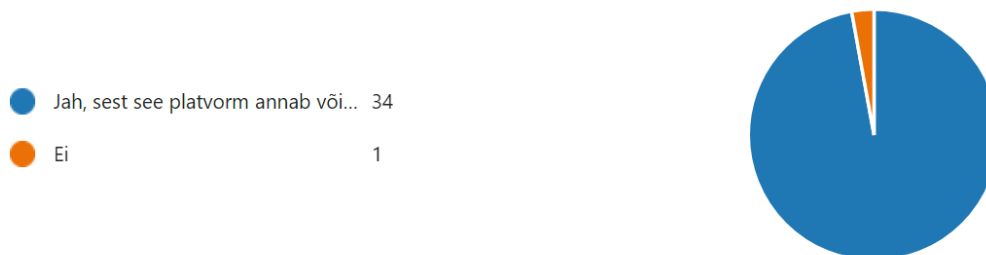


Joonis 4. Vastajate SQL-oskuse enesehinnang.

Pealehe keskmine hinne oli 4.3. Kiideti lehe selget ja kasutajasõbralikku kujundust ning mitmekülgseid valikuid, kuid mõned vastajad leidsid, et lehel on liiga palju teksti. Ülesannete ja kogumi loendi lehe keskmine hinne oli 4.4. Positiivset tagasisidet said lehe selgus ja intuiitiivsus, kuid mõned vastajad soovisid paremaid ülesannete nimesid ja lühikirjeldusi (märkus – see pole kasutajaliidese küsimus). Ülesande lahendamise lehe keskmine hinne oli 4.5. Kiideti lehe funktsionaalsust ja kasutajasõbralikkust. Ülesannete ja ülesannete ajaloo lehe keskmine hinne oli samuti 4.5. Enamik kasutajaid pidas lehti kasulikuks, kuna need võimaldavad vaadata varasemaid lahenduskatseid ja analüüsida vigu, kuid mõned ei leidnud sellistest lehtedest palju kasu, kuigi hindasid visuaalset kujundust. "Statistika" lehe keskmine hinne oli 4.5. Vastajad kiitsid graafilist esitlust ja üldist ülevaadet, kuid mõned soovisid rohkem detaile ja paremat värvide kasutust. Veebirakenduse uuringu tulemustel põhinevaid tulpdiagramme iga lehe kohta saab vaadata lisast 9.

Tulevase kasutamise osas väljendas 34 inimest 35-st positiivset suhtumist platvormi kasutamisse tulevikus, eriti „Andmebaasi I“ kursuse kontekstis (vt joonis 5). Üks kasutaja teatas, et ei kasutaks platvormi tulevikus, kuid põhjuseid ei ole andmestikus täpsustatud.

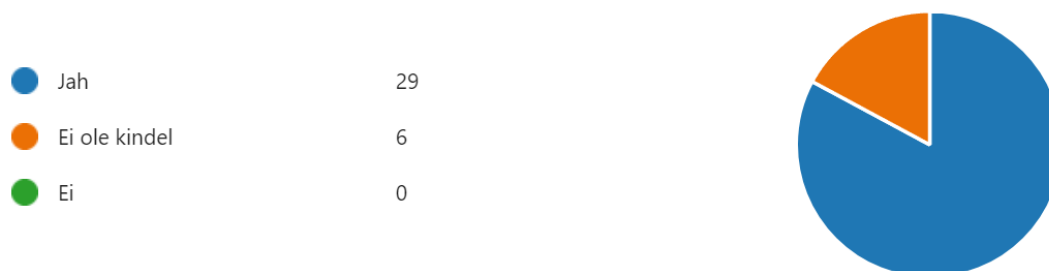
14. Kas Te kasutaksite meie platvormi tulevikus, näiteks "Andmebaasi I" kursuse raames?



Joonis 5. Sektordiagramm tulevase kasutamise kohta

Platvormi soovitamise osas soovitaksid 29 kasutajat seda oma kolleegidele või sõpradele, kuid kuus kasutajat ei olnud kindlad, kas nad soovitaksid platvormi teistele, kuna nad jäid neutraalseks (vt joonis 6).

16. Kas Te soovitaksite meie platvormi kolleegidele või sõpradele?



Joonis 6. Sektordiagramm platvormi soovitamise kohta.

5.3.3 Tulemuste analüüs

Küsitluse tulemused viitavad üldisele rahulolule platvormiga. Õpilaste keskmine hinnang oma SQL-oskustele (3) näitab, et „Andmebaasid I“ kursuses aitab neil oma oskuseid parandada. Kursuse algul prototüübi kohta kogutud tagasisides oli vastajate keskmine hinnang oma SQL oskusele madalam. Kõrged keskmised hinnangud erinevatele lehtedele (4.3 – 4.5) näitavad, et platvormi erinevad võeti hästi vastu.

Pealehe kasutajate kommentaarid varieerusid, kuid üldiselt oldi esilehega rahul. Paljud leidsid, et leht on selgelt üles ehitatud ja lihtne kasutada ning sisselogimise ja testide alustamise nupud on hästi leitavad. Tekst on küll kergesti loetav, kuid teisalt on seda palju ja see ei ole hästi hoomatav. Teksti jaotamine ja visuaalse disaini uuendamine aitaksid muuta lehe informatiivsemaks ja meeldivamaks. Kokkuvõttes on esileht lihtne ja funktsionaalne, kuid vajab visuaalset täiustamist ja paremat tekstijaotust, et parandada kasutajakogemust.

Ülesannete ja kogumi loendi lehe korral kiitsid vastajad lehe värvikasutust, mis aitab ülesandeid kiiremini eristada. Lehe loogiline ülesehitus ja lihtne navigatsioon said samuti head tagasisidet. Üks kommentaar tõi esile lehe arusaadavuse ja intuiivsuse ning funktsioonide head toimimist: „Kõik tundus loogiline, navigatsioon on lihtne.“ Samas leiti, et ülesannete nimetused on veidrased ja vajaksid konkreetsemaid nimesid. Selle kohta tuleb öelda, et tegemist pole kasutajaliidese, vaid esitatava informatsiooni probleemiga. Üldiselt peeti lehte väga heaks, kuna värvid aitavad kiiremini eristada, mis on tehtud või tegemata. Lisaks sooviti edaspidi rohkem ülesandeid. See on jällegi esitatava informatsiooni, mitte kasutajaliidese, küsimus. Kokkuvõttes võeti leht hästi vastu, kuid vajab ülesannete nimetuste täpsustamist ja ülesannete arvu suurendamist.

Kasutajate tagasiside põhjal on ülesannete lahendamise lehe funktsionaalsus suurepärane, kuid väljanägemine võiks olla parem. Kasutajad kiitsid, et vastuse õigsus on kohe näha „Väga hea, et on kohe näha, kas vastus on õige või vale.“ ning oodatud tulemuse nägemine aitab lahendamisele kaasa. Lahenduse kirjutamise koha ja ülesande teksti ekraani erinevatele ekraani pooltele jaotamine muudab ülesande lahendamise mugavaks. Samuti hinnati kõrgelt vihjete küsimise võimalust. Negatiivse poole pealt leiti, et ülesande sisust oli vahepeal keeruline aru saada, eriti tabelite ja veergude nimed olid segased (see on esitatava info küsimus). Samuti tundsid kasutajad puudust nupust „Järgmine ülesanne sellest kogumist“. Katsetatud versioonis pidi minema tagasi ülesannete loendi juurde ja valima sealt vajaliku kogumi ning selle uue ülesande, mis ei olnud kasutajasõbralik.

Mis puudutab „Ülesannete ajalugu“ lehe funktsiooni, siis mõned vastasid, et ei hakkaks seda tihti kasutama, kuid on hea, et see on olemas. Samuti leiti, et kogu vajalik info on olemas ja kergesti loetav: „Kõik vajalik olemas ja kergesti loetav, meeldib väga.“ Lehekülg võimaldas saada efektiivselt ülevaate enda sooritustest ning leida üles need

ülesanded, mis vajaksid veel harjutamist, ehk need, mille läbimiseks kulus algul rohkem kui üks katse. Üks kasutaja mainis: „Hea võimalus näha täpsemalt ülesande lahendamise ajalugu. Kujundus on selge, mugav ja kergesti mõistetav.“

Kuigi tundus, et suurt vajadust „Ülesannete ajalugu“ lehe järele ei ole, hindasid vastajad selle siiski heaks, sest see annab võimaluse näha igat ülesande esitust, mis võib aidata õppijatel analüüsida, mida nad algul valesti tegid. Visuaalselt peeti lehte väga ilusaks ning kõik nupud olid loogiliselt paigutatud. Kokkuvõttes hinnati lehe funktsionaalsust ja kujundust kõrgelt, kuigi selle kasutamise sagedus kujuneb madalaks.

Kasutajate arvates on statistika leht kasulik oma tulemuste analüüsimiseks. Positiivse poole pealt toodi esile protsentide graafiline kujutamine: „Meeldis, et protsendid on graafiliselt kujutatud, nii on arengut paremini näha.“ Üldiselt peeti lehte heaks ülevaateks, kuid kasutajad arvasid, et seda saaks visuaalselt ligitõmbavamaks teha. Kuigi leht ei ole halb, oli kasutajatel tunne, et selle väljanägemist võiks visuaalselt veelgi parandada, kuid täpseid ettepanekuid selle kohta ei osatud anda.

Juhendaja (ja ka süsteemi haldaja) hinnangul andis küsitlus palju kasulikku infot ka selle kohta, kuidas ülesandeid sõnastada.

Küsitlusele vastasid enamasti üliõpilased, kes olid semestri jooksu ka muidu aktiivselt SQLi õppinud ning selles häid tulemusi saanud. Seega ei saadud tagasisidet nendelt, kellel oli SQLi õppimisega raskuseid.

5.3.4 Tulemuste analüüsi alusel tehtud muudatused

Pärast küsitlust tehti mitmeid parandusi ja täiendusi, et kasutajakogemus oleks sujuvam ja mugavam. Uuendati veergude ja tabelite nimesid, et need oleksid selgemad ja üheselt arusaadavad. Lisati nupp „Tagasi sama kogumi juurde“, mis võimaldab kasutajatel liikuda otse kogumi juurde ja valida samast kogumist teisi ülesandeid. Parandati kalendri funktsionaalsust ja nähtavust ajaloo lehel, et kasutajad saaksid valida ka eelnevaid kuid ning aastaarv oleks visuaalselt hästi nähtav. Alguse ja lõpu aja valikul on vaikimisi kellaajad 00:00 ja 23:59.

Ajaloo lehel on nüüd võimalik paremini filtreerida lahendamise seisundi järgi. Eemaldati valik „Pole proovitud“, kuna seal lehel ei esitata proovimata ülesandeid. Lisati ka filtrite puhastamise ja kõigi katsete kuvamise nupp. Ülesannete ja ajaloo lehtedel muutub rida

kursoriga selle kohale minnes ning ülesandele viib mistahes rea väljale klõpsates, mitte ainult nimele. Seisundi ja keerukuse väärtustel on erinevad värvid, et kasutajate tähelepanu ei hajuks. Pikad ülesande nimed murduvad nüüd erinevatele ridadele, mis võimaldab sisu haldajal soovi korral kasutada ka pikemaid ülesande nimesid.

Ülesande lehel on nüüd tabelite struktuuri pilt kohe avatud. Statistika lehel on edukate ja ebaedukate katsetuste protsendid sulgudes (eelnev esitus tekitas kasutajates segadust, et kas kuvatakse protsentide vahemikku). Ajaloo lehel on otsimise ja ajaloo tühjendamise nupud üksteise alla joondatud. Statistika lehel on nähtav kulutatud aeg. Konkreetse katse andmeid vaadates näeb, kui palju aega oli ülesande lahendamiseks kulunud. On võimalik liikuda järgmise kogumi ülesande juurde, mis ilmub siis, kui ülesanne on lahendatud. Ülesande lehel on märge PostgreSQL'i kasutamise kohta. Varasemate katsete vaatamisel on katsetuste andmed samal lehel ja valitud katset näidatakse viimase lahenduse kohal, mitte eraldi aknas.

Lisaks parandati keskkonnas grammatilisi vigu. Selle kohta tuleb öelda, et kuna kasutajaliidese tekstid on eraldi keelefailides, siis taolisi vigu on lihtne parandada ka hiljem.

5.4 Töö tegemise refleksioon

Selles jaotises analüüsitakse tööd selles vaates, et mis läks töö tegemise juures hästi, mis läks halvasti ja miks see juhtus ning kellele ja miks oleks antud tööst kasu.

5.4.1 Asjad, mis läksid hästi

Selle töö käigus viidi läbi mitmeid edukaid tegevusi. Esmalt analüüsiti enne uue süsteemi arendamise alustamist põhjalikult teisi sarnaseid süsteeme ja uuriti ka teaduskirjandust. See analüüs võimaldas tuvastada ja õppida varasematest vigadest ning tõi kaasa uute ja paremate funktsioonide lisamise loodud platvormile. Teiseks oli arenduse planeerimine meetoodiline ja hästi struktureeritud, hõlmates muuhulgas kasutajaliidese prototüübi loomist, tarkvara põhjalikku testimist ja tarkvara lõplikku juurutamist. Selline läbimõeldud aja planeerimine tagas sujuva töövoogu.

Kasutajate küsitluste tulemused näitasid, et kasutajad olid süsteemi testimisse aktiivselt kaasatud, ning vastused näitasid nende rahulolu ja süsteemi vastuvõetavust. Projekti

õnnestumise oluline tegur oli ka meeskonna kõrge koostöövõime. Kiire omavaheline suhtlus ja võrdne ülesannete jagamine meeskonnas tagasid tõhusa tööprotsessi. Paaristöö võimaldas jagada ülesandeid, aidata üksteist, jagada kogemusi ja leida vigu, mis parandasid üldist töö kvaliteeti ja efektiivsust.

5.4.2 Asjad, mis läksid halvasti

Lõputöö tegemisel ilmnisid mitmed mitte kriitilised probleemid, mis mõjutasid projekti edukust. Esiteks, projekti alustamine hilines ning lõputööga tegelemine viibis, mis tõi kaasa ajapuuduse ja täiendava funktsionaalsuse puuduliku realiseerimise. Aja vale hindamine tekitas olukorra, kus planeeritud täiendusi ei olnud võimalik õigeaegselt ellu viia.

Teiseks, projekti käigus ilmnisid puudused nii praktilistes oskustes kui ka teoreetilistes teadmistes PHP keeles programmeerimise kohta. See tõi kaasa vajaduse omandada olulisi oskusi ja teadmisi töö käigus, mis omakorda mõjutas arendustöö efektiivsust ja projekti üldist kvaliteeti.

5.4.3 Asjad, mida teha tööd korrates teistmoodi

Tulevikus sarnaseid projekte tehes tuleks arvestada, et lõputööga on vaja alustada varakult, et oleks piisavalt aega nii arendustöödeks kui ka valminud projekti põhjal lõputöö dokumendi kirjutamiseks. Varajane alustamine võimaldaks vältida ajapuudusest tingitud probleeme ja tagada, et kõik planeeritud funktsioonid saaksid õigeaegselt realiseeritud.

5.4.4 Tehtud töö kasulikkus

Tehtud töö kasulikkus seisneb selles, et valminud tarkvara saab kasutada SQL-i õppimisel ja praktiseerimisel. See keskkond on plaanis võtta kasutusele Tallinna Tehnikaülikooli andmebaaside kursustes. See keskkond võimaldab harjutada ülesandeid erinevat tüüpi SQL lausete koostamise kohta. Süsteemi kasutajad saaksid järjepidevalt SQL lausete kirjutamist harjutada. Lisaks on sellel lihtne ja arusaadav kasutajaliides ning võimalus vaadata statistilisi andmeid ning küsida tagasisidet õppejõu käest või anda talle märku ülesandega seotud probleemidest. Töö autorid ei leidnud sarnaseid keskkondi, mis pakuksid kõiki selliseid võimalusi.

Töö esitamise ajal on SQL lausete iseseisva harjutamise veebikeskkonnale ligipääs Tallinna Tehnikaülikooli andmebaaside õppeainete üliõpilastel. See pole halb, kuna neil on võimalus tasuta (kuna paljud analoogid on tasulised) ja piiramatus mahus oma oskuseid iseseisvalt kasvatada.

Osapooled, kellele on töö tulemustest kasu

- Tallinna Tehnikaülikooli üliõpilased: Üliõpilastel on võimalus täiendada oma SQLi oskusi iseseisvalt ja tasuta, mis parandab nende õpikogemust ja oskuseid.
- Õppejõud: Õppejõud saavad pakkuda üliõpilastele lisavõimalusi praktiseerimiseks. Kuna süsteem logib ülesannete lahendamise katseid, siis on võimalik saada ülevaadet üliõpilaste edusammudest statistika kaudu või nende tehtavatest tüüpilistest vigadest. Keskkonda saaks soovi korral kasutada ka selleks, et osaliselt automatiseerida kursuse kohustuslike või vabatahtlike ülesannete kontrollimist ja seega kaasata seda keskkonda ka hindamisse.
- Tallinna Tehnikaülikool: Ülikool saab pakkuda innovatiivset ja tasuta õppimisvõimalust, mis võib tõsta nende mainet ja atraktiivsust tulevaste üliõpilaste seas.

Äriline kasu

Antud töö tulemusel on ärilist (rahaliselt mõõdetavat) kasu mitmel viisil.

- Üliõpilaste parem ettevalmistus: Paremad SQLi oskused võivad viia kõrgema tööalase konkurentsivõimeni, mis võib tõsta ülikooli lõpetajate keskmist palka ja seeläbi ülikooli mainet.
- Keskkonna edasine levitamine: Tulevikus võib kaaluda keskkonna pakkumist teistele ülikoolidele või muudele organisatsioonidele, mis võimaldaks selle müüki või litsentseerimist ja seeläbi tulu teenimist.
- Kõigile huvilistele kättesaadavaks tegemine: Keskkonda võib pakkuda kõikidele huvilistele tasuta eest, mis avardaks kasutajaskonda ja suurendaks otsest rahalist tulu. Tasuta pakkumine suurendaks samuti kasutajaskonda ning tõstaks ülikooli mainet.

- Monetiseerimine: Kui minna keskkonna pakkumisele väljapoole ülikooli, siis võib keskkonda edasi arendada ja pakkuda täiendavaid tasulisi funktsioone, mis toovad otsest rahalist tulu. Näiteks võiks lisada algoritmi, mis analüüsib kasutaja peamisi nõrkusi ja annab neile soovitusi, kuhu edasi liikuda, et oma oskuseid parandada.

5.5 Edasine arendus

Käesolevas arendustsüklis on leitud mitmeid süsteemi täiustusvõimalusi, et parandada kasutajakogemust ja suurendada süsteemi funktsionaalsust. Paljud nendest mõtetest pärinevad kasutajate tagasisidest.

Esiteks tuleks sisselogimisel realiseerida parooli meeldejätmise, et kasutajad saaksid valida, kas nad soovivad oma parooli meelde jätta, et järgmine kord sisse logides ei peaks seda uuesti sisestama. Samuti tuleks realiseerida ülesannete nimekirjas filtrite ja sorteerimise vaikute meeldejätmise võimalus, et kasutajate valikud salvestataks ja nad ei peaks neid iga kord uuesti tegema. Lisaks oleks hea luua automaatne poolelioleva koodi salvestamise funktsionaalsus, mis tagab, et kasutajad ei kaotaks oma tööd, kui nad ülesandest väljuvad või veebilehitseja sulgevad.

Süsteemi üheks täienduseks oleks harjutajatele ülesannete lahendamise edetabelite vaatamise võimaldamine. Nii saaksid nad ennast võrrelda teistega ning võiks tekkida võrdlusmoment ja hasart ülesandeid lahendada.

Samuti oleks võimalikuks oluliseks täienduseks veebilehest mobiilisõbraliku versiooni loomine. Praeguses versioonis on loodud kasutajaliides mõeldud kasutamiseks töölaua arvutis.

Kasulikuks täienduseks oleks ka lahendus, mis pakub kasutajatele vihjeid analüüsides nende esitatud vastust ja pakkudes asjakohaseid soovitusi tulenevalt vigadest, mida kasutaja konkreetses lahenduses tegi. Lisaks võiks süsteemi lisada funktsionaalsuse, mis pakub kasutajatele viiteid ja teemakohaseid materjale, et toetada nende õppimist ja ülesannete lahendamist.

Edaspidi võiks keskkond hakata toetama täiendavaid andmebaasisüsteeme nagu MySQL, Oracle ja MS SQL Server, et suurendada süsteemi kasulikkust ja kasutusvõimalusi.

Veel üheks täienduseks oleks veebipõhine administraatori kasutajaliides, mis võimaldab administraatoritel jälgida ja analüüsida üliõpilaste ülesannete täitmise statistikat, lisada uusi ülesandeid ning süsteemi tõhusalt hallata. Praegu toimub see läbi administraatori arvutis oleva programmi.

6 Kokkuvõte

Käesoleva töö eesmärk oli luua veebipõhine keskkond SQL-i iseseisvaks harjutamiseks, mis vastab Tallinna Tehnikaülikooli andmebaaside kursuste vajadustele. Töö käigus analüüsiti olemasolevaid lahendusi, tehti kindlaks uue süsteemi funktsionaalsed ja mittefunktsionaalsed nõuded ning arendati platvorm, mis toetab mitmekesiste SQL lausete harjutamist.

Töö tulemusel valmis PHP keeles kirjutatud veebirakendus, mis pakub Tallinna Tehnikaülikooli üliõpilastele võimalust iseseisvalt harjutada SQL lauseid PostgreSQL andmebaasisüsteemi põhjal. Platvorm võimaldab erinevate ülesannete lahendamist (sh nii andmete otsimise, andmete muutmise kui ka andmestruktuuride loomise muutmise ja kustutamise ülesanded), automaatset tagasiside saamist ja lahendamise vihjete kasutamist. Lisaks pakub keskkond kasutajatele võimalust jälgida oma edusamme ja analüüsida lahendatud ülesandeid.

Töö tegemise käigus küsiti süsteemi võimalikelt kasutajatelt korduvalt tagasisidet ning saadud vastused aitasid platvormi paremaks muuta. Töö autorid tänavad kõiki tagasiside andjaid!

Lõppkokkuvõttes võiks loodud veebirakendus aidata kaasa SQL-i oskuste omandamisele ja süvendamisele, pakkudes üliõpilastele ja teistele huvilistele kasutajasõbralikku ja tõhusat tööriista, mis toetab nende akadeemilist ja professionaalset arengut.

Kasutatud kirjandus

- [1] Modern SQL, „A lot has changes since SQL-92”. [Võrgumaterjal]. Available: <https://modern-sql.com/> [Kasutatud 07 05 2024].
- [2] Institute of Electrical and Electronics Engineers, „The Top Programming Languages 2023”. [Võrgumaterjal]. Available: <https://spectrum.ieee.org/the-top-programming-languages-2023> [Kasutatud 23 02 2024].
- [3] S. Tupailo, „General methods in inductive program synthesis”, 1996.
- [4] M.K. Sein, O. Henfridsson, S. Purao, M. Rossi, R. Lindgren, „Action design research”, MIS Quart. 35, 37-56, 2011.
- [5] K. Petersen¹, C. Wohlin¹, ja D. Baca, „The Waterfall Model in Large-Scale Development”, 2009.
- [6] HackerRank, „veebirakendus SQL päringute kirjutamiseks ja käivitamiseks”. [Võrgumaterjal]. Available: <https://www.hackerrank.com/> [Kasutatud 23 02 2024].
- [7] LeetCode, „veebirakendus SQL päringute kirjutamiseks ja käivitamiseks”. [Võrgumaterjal]. Available: <https://leetcode.com/problemset/database/> [Kasutatud 23 02 2024].
- [8] PostgreSQL Exercises, „PostgreSQLi ja selle kaudu ka SQLi interaktiivne harjutamine”. [Võrgumaterjal]. Available: <https://pgexercises.com/> [Kasutatud 23 02 2024].
- [9] SQLPad, „veebirakendus SQL päringute kirjutamiseks ja käivitamiseks”. [Võrgumaterjal]. Available: <https://sqlpad.io/> [Kasutatud 23 02 2024].
- [10] J.E. Laamann, „Interaktiivse veebiõpiku loomine ainele "Sissejuhatus andmebaasidesse", Bakalaureusetöö, Tartu Ülikool, 2019. [Võrgumaterjal]. Available: <https://dspace.ut.ee/items/129b8986-afef-4c23-a8e7-56ddc88334>
- [11] A. Ahadi, J. Prior, V. Behbood, R. Lister, „Students' Semantic Mistakes in Writing Seven Different Types of SQL Queries”, pp. 272–277, 2016, doi: 10.1145/2899415.2899464.
- [12] A. Ahadi, J. Prior, V. Behbood, A. Vihavainen, R. Lister, „Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and its Application to Predicting Students' Success”, pp. 401-406, 2016, doi: 10.1145/2839509.2844640.
- [13] M. Fabijanić, G. Đambić, B. Fulanović, „A Novel System for Automatic, Configurable and Partial Assessment of Student SQL Queries”, 2020, doi: 10.23919/MIPRO48935.2020.9245264.
- [14] S. Nayak, R. Agarwal, S. Kumar Khatri, „Review of Automated Assessment Tools for grading student SQL queries”, 2022, doi: 10.1109/ICCCI54379.2022.9740799.
- [15] Andmekaitse inspeksioon, „Nõuandeid veebi ligipääsetavaks muutmiseks”. [Võrgumaterjal]. Available: <https://www.aki.ee/uudised/nouandeid-veebi-ligipaasetavaks-muutmiseks/> [Kasutatud 18 05 2024].
- [16] Epikoda, „Veebide ligipääsetavus”. [Võrgumaterjal]. Available: <https://epikoda.ee/mida-me-teeme/ligipaasetavus/veebide-ligipaasetavus> [Kasutatud 18 05 2024].
- [17] W3C, „Markup Validation Service”. [Võrgumaterjal]. Available: <https://validator.w3.org/> [Kasutatud 18 05 2024].

- [18] WebAIM, „Contrast Checker”. [Võrgumaterjal]. Available: <https://webaim.org/resources/contrastchecker/> [Kasutatud 18 05 2024].
- [19] PHP, „PHP Data Objects manual”. [Võrgumaterjal]. Available: <https://www.php.net/manual/en/intro.pdo.php/> [Kasutatud 11 05 2024].
- [20] CodeMirror, „Extensible Code Editor”. [Võrgumaterjal]. Available: <https://codemirror.net/> [Kasutatud 18 05 2024].
- [21] SQL-formatter, „SQL-koodi vormindamine”. [Võrgumaterjal]. Available: <https://unpkg.com/browse/sql-formatter@2.3.3/> [Kasutatud 18 05 2024].
- [22] Flatpickr, „Kalendri raamistik”. [Võrgumaterjal]. Available: <https://flatpickr.js.org/> [Kasutatud 18 05 2024].
- [23] Lightgallery, „Pildi galerii plugin”. [Võrgumaterjal]. Available: <https://www.lightgalleryjs.com/> [Kasutatud 18 05 2024].
- [24] R. Ramler, C. Klammer „A Journey from Manual Testing to Automated Test Generation in an Industry Project”, IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2017.
- [25] zyBooks, „Interaktiivne õpik”. [Võrgumaterjal]. Available: <https://www.zybooks.com/> [Kasutatud 26 02 2024].
- [26] SQLZoo, „Interaktiivne SQL õpik”. [Võrgumaterjal]. Available: https://sqlzoo.net/wiki/SQL_Tutorial [Kasutatud 23 02 2024].
- [27] StrataScratch, „veebirakendus SQL päringute kirjutamiseks ja käivitamiseks”. [Võrgumaterjal]. Available: <https://www.stratascratch.com/> [Kasutatud 23 02 2024].
- [28] DataLemur, „veebirakendus SQL päringute kirjutamiseks ja käivitamiseks” [Võrgumaterjal]. Available: <https://datalemur.com/> [Kasutatud 23 02 2024].
- [29] W3resource, „veebirakendus SQL päringute kirjutamiseks ja käivitamiseks”. [Võrgumaterjal]. Available: <https://www.w3resource.com/sql-exercises/> [Kasutatud 23 02 2024].
- [30] Mode Analytics, „veebirakendus SQL päringute kirjutamiseks ja käivitamiseks”. [Võrgumaterjal]. Available: <https://mode.com/sql-tutorial> [Kasutatud 23 02 2024].
- [31] V. Suprun ja L. Demidov, „Kasutajaliidese prototüübi Figma”. [Võrgumaterjal]. Available: <https://figma.fun/d4S6po> [Kasutatud 23 02 2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Meie, Leonid Demidov ja Vladislav Suprun

1. Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “SQL ülesannete iseseisva harjutamise veebikeskkonna loomine”, mille juhendaja on Erki Eessaar
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Oleme teadlikud, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autoritele.
3. Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

20.05.2024

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Lõpp-punktide nimekiri

Tabel 3. Lõpp-punktide nimekiri.

Meetod	Teekond	Kirjeldus
PUT	/api/locales/:localeId	Määra kasutaja saidi keel keele id põhjal.
GET	/api/locales	Saab saidi saadaolevate keelte loendi.
POST	/api/users/auth	Autendib kasutajat.
POST	/api/users/logout	Logib tarkvarast välja.
GET	/api/collections	Saab ülesannete kogumite loendi.
GET	/api/collections/:collectionId	Saab kindla ülesannete kogumi andmed selle id põhjal.
GET	/api/tasks	Saab kõikide ülesannete loendi.
GET	/api/tasks/userStateTypes	Saab ülesannete olekute loendi.
GET	/api/tasks/statementTypes	Saab ülesannete tüüpide (SQL lause tüüpide) loendi.
GET	/api/tasks/difficultyLevels	Saab ülesannete raskusastmete loendi.
GET	/api/tasks/databases	Saab ülesannete andmebaaside loendi.
GET	/api/tasks/random	Saab juhusliku ülesande id.
GET	/api/tasks/:taskId	Saab ülesande täielikud andmed vastavalt selle id-le (kirjeldus, nimi, raskusaste, tüüp, kasutaja olek jne)
GET	/api/tasks/:taskId/expectedResult	Saab oodatud tulemused vastavalt ülesande id-le.
GET	/api/tasks/:taskId/tables	Saab ülesande tabelite loendi vastavalt selle id-le.
GET	/api/tasks/:taskId/tables/:tableId	Saab ülesande tabeli andmed vastavalt selle id-le.
GET	/api/tasks/:taskId/nextHint/:lastHintId	Saab järgmise vihje vastavalt viimasele päringule.
GET	/api/tasks/:taskId/allHints	Saab kõik vihjed vastavalt ülesande id-le.

Meetod	Teekond	Kirjeldus
GET	/api/tasks/:taskId/nextSolution/:lastSolutionId	Saab järgmise võimaliku lahenduse vastavalt viimasele päringule.
GET	/api/tasks/:taskId/allSolutions	Saab kõik võimalikud lahendused vastavalt ülesande id-le.
POST	/api/tasks/:taskId/submitTask	Saadab ülesande lahenduse kontrollimiseks ja tagasiside saamiseks.
GET	/api/tasks/:taskId/attempts	Saab ülesande katsete loendi vastavalt selle id-le.
GET	/api/tasks/:taskId/attempts/last/full	Saab üksikasjalikud andmed viimase katse kohta vastavalt selle id-le.
GET	/api/tasks/:taskId/attempts/last/basic	Saab põhilised andmed viimase katse kohta vastavalt selle id-le.
GET	/api/tasks/:taskId/feedbacks	Saab tagasiside ülesande kohta vastavalt selle id-le.
POST	/api/tasks/:taskId/feedbacks	Saadab tagasiside ülesande kohta.
GET	/api/attempts	Saab katsete loendi.
GET	/api/attempts/:attemptId	Saab katse andmed vastavalt selle id-le.
PUT	/api/attempts/clean	Puhastab kasutaja katsete ajaloo.
GET	/api/statistics	Saab statistika põhiaandmed lahenduste kohta.
GET	/api/statistics/all	Saab kokkuvõtte kõikide katsete statistika kohta.

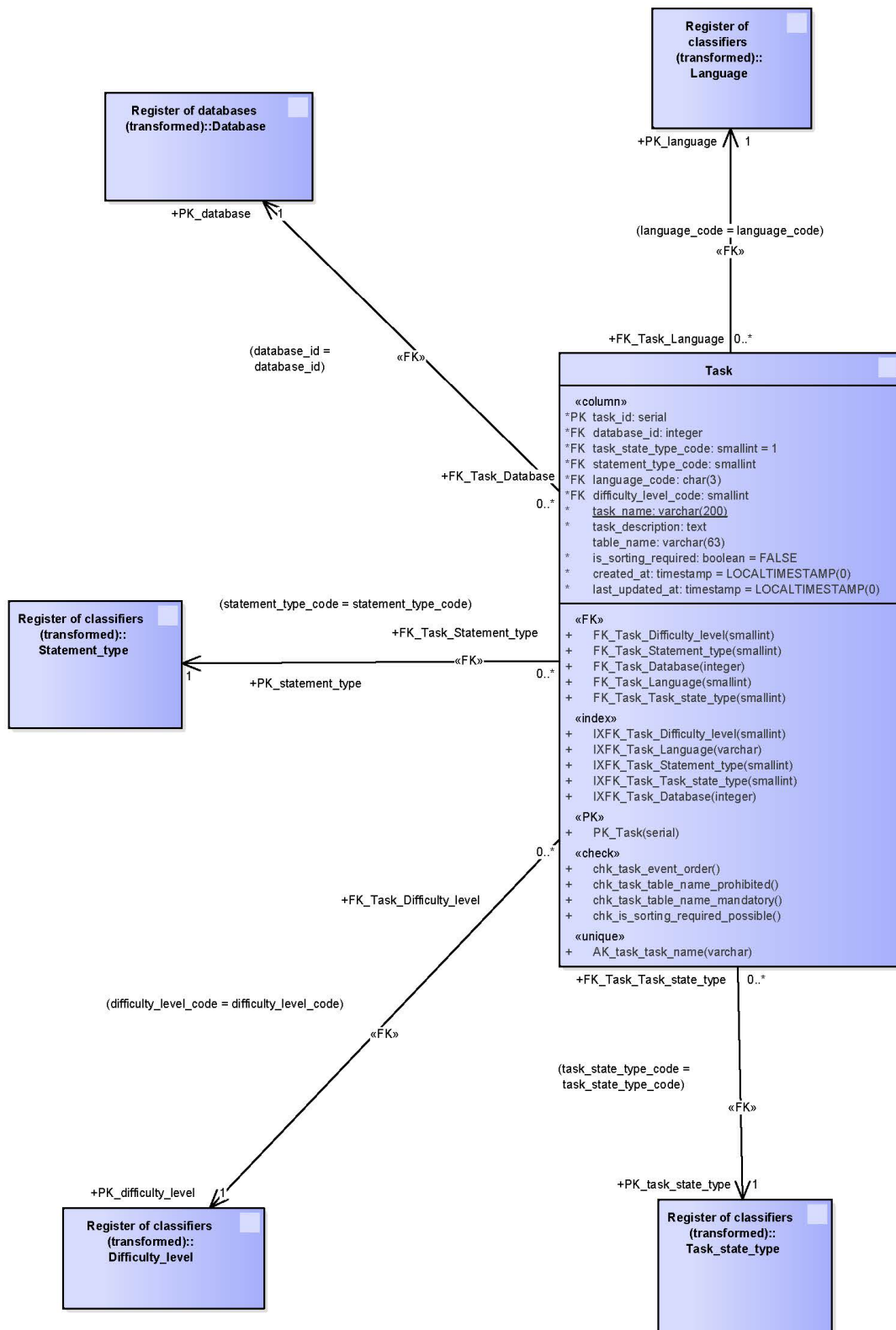
Lisa 3 – Võimalikud vead kasutaja koodi kontrollimisel

Tabel 4. Võimalikud vead kasutaja koodi kontrollimisel.

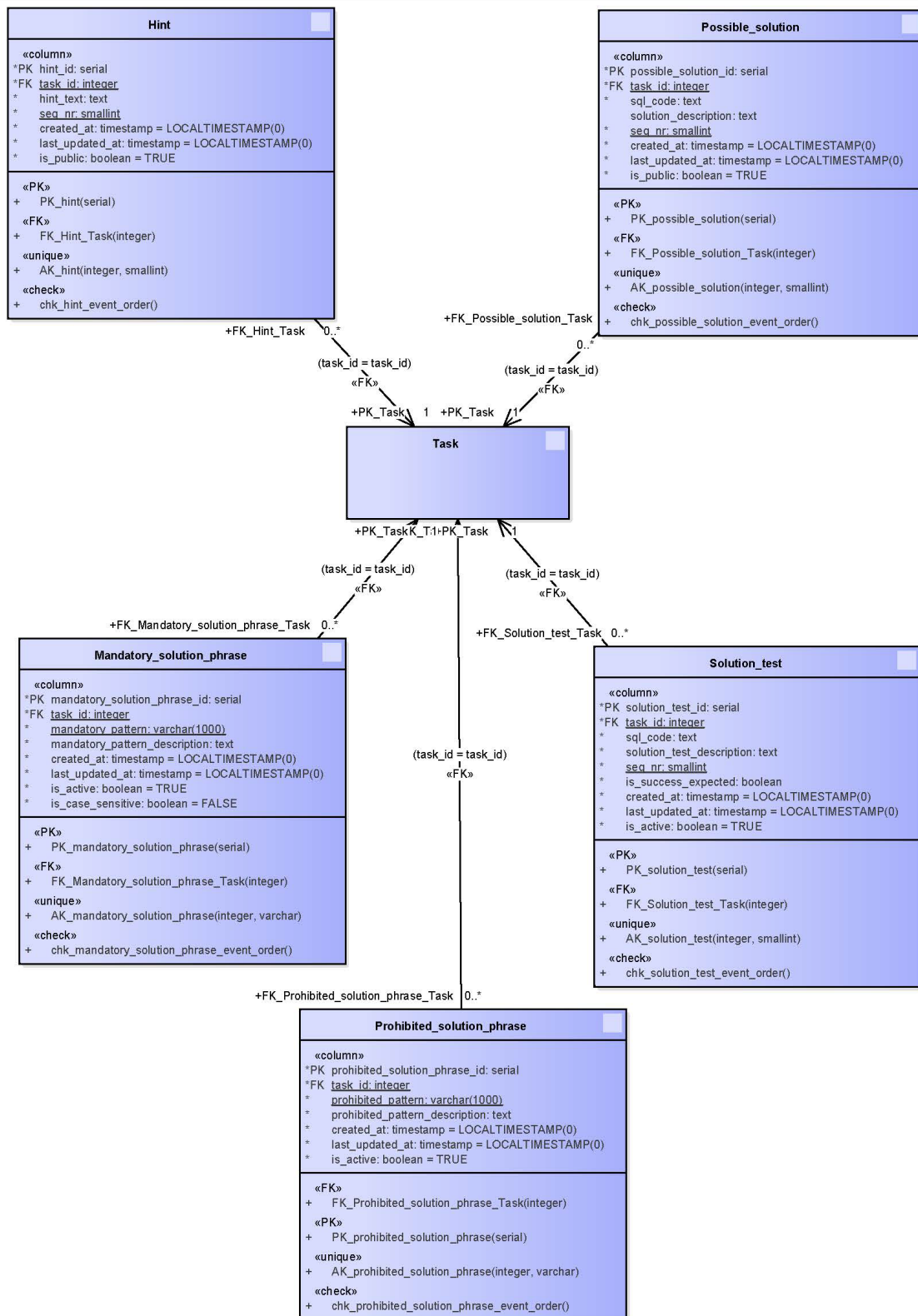
Kood	Nimi	Kirjeldus	Kommentaar
1	Lause ei vasta ülesande tüübile	Lauset ei saa käivitada!	Näiteks SELECT lause asemel DELETE
2	Lauses üldine mittelubatud muster	Lauset ei saa käivitada!	Näiteks lauses -- või DROP DATABASE
3	Lause ei vasta kõigile ülesande-spetsiifilistele nõutud mustritele	Lauset ei saa käivitada!	Näiteks on nõutud FETCH FIRST n ROWS WITH TIES
4	Lauses ülesande-spetsiifiline mittelubatud muster	Lauset ei saa käivitada!	Näiteks pole lubatud kasutada FETCH FIRST n ROWS WITH TIES
5	Lause pole õige tabeli põhjal	INSERT, UPDATE, DELETE ja MERGE lausete korral. Lauset ei saa käivitada!	
6	Lauses puudub nõutud sorteerimine	SELECT lause korral. Lauset ei ole põhjust käivitada!	
7	Käivitamine ebaõnnestus		Näitab PHP vigu serveris, kui neid on
8	Tulemuses nõutust rohkem veerge	SELECT lause korral	
9	Tulemuses nõutust vähem veerge	SELECT lause korral	
10	Tulemuses ebaõige veergude järjekord	SELECT lause korral	
11	Tulemuses ebaõiged veerud	SELECT lause korral	
12	Tulemuses veergude nimed nõutust liiga erinevad	SELECT lause korral	Nimede Levenshteini kaugus e tähemuudatuste arv, et saada teine nimi, on suurem kui 2 (hindamiseks kasutatakse PostgreSQL'i laiendusse

Kood	Nimi	Kirjeldus	Kommentaar
			<i>fuzzystmatch</i> kuuluvat rutiini).
13	Valed leitud andmed	SELECT lause korral	
14	Tulemus pole sorteeritud nii nagu vaja	SELECT lause korral	
15	Vale andmemuudatus	INSERT, UPDATE, DELETE ja MERGE lausete korral	
16	Ebaõnnestunud test	CREATE TABLE, ALTER ja DROP lausete korral	Näitab iga testi kirjeldust, kui see on saadaval

Lisa 4 – Ülesannete registre andmebaasi disaini skeemid

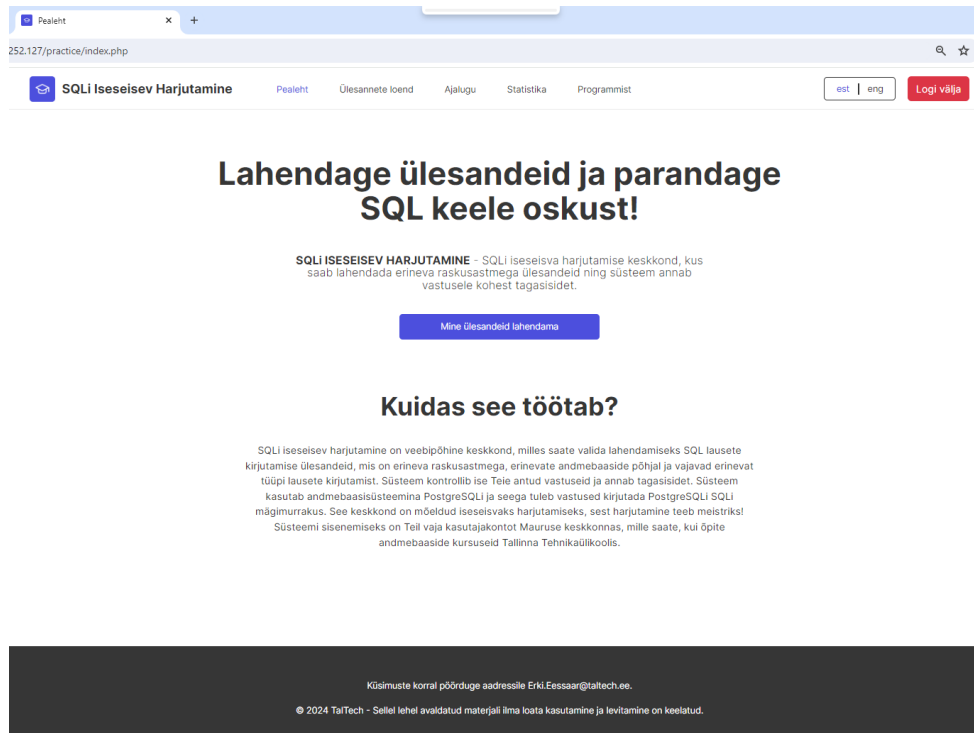


Joonis 7. Ülesannete registre andmebaasi disaini skeem Enterprise Architect modelleerimisvahendis.

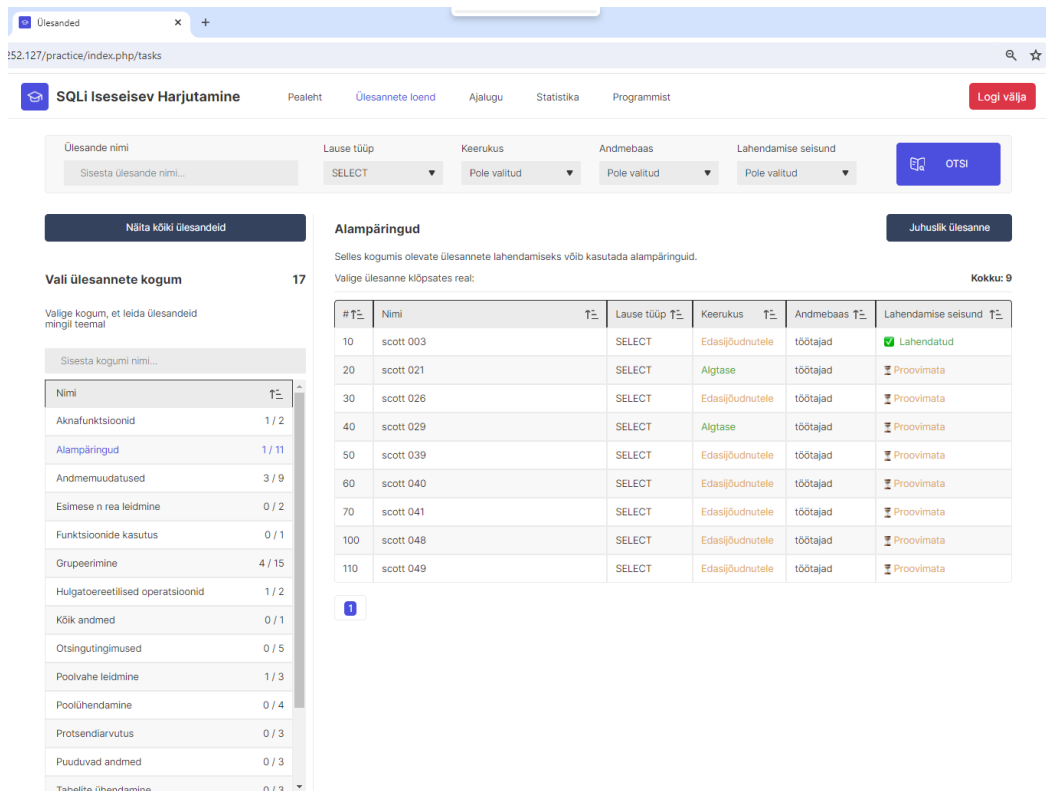


Joonis 8. Ülesannete register andmebaasi disaini skeem Enterprise Architect modelleerimisvahendis.

Lisa 5 – Kasutajaliidese illustatsioonid



Joonis 9. Pealeht.



Joonis 10. Leht ülesannete loendiga.

scott 003 x +

!52.127/practice/index.php/tasks/3?collection_id=12

SQLi Iseseisev Harjutamine Pealeht Ülesannete loend Ajalugu Statistika Programmist Logi välja

Tagasi "Alampäringud" juurde Järgmise ülesande juurde →

Ülesande nimi: scott 003 Lause tüüp: SELECT Andmebaas: töötajad
Keerukus: Edasijõudnutele Lahendamise seisund: Lahendatud LAHENDAMISE AJALUGU

Ülesanne: [Saada teade](#)
Ülesande kirjeldus: Vaata oodatud tulemust ▲
Leia kõik osakonnad, kus ei tööta ühtegi töötajat. Väljastage osakonna kood ja nimetus. Kasutage selleks EXISTS predikaati koos korreleeruva alampäringuga. Sorteerige tulemus osakonna numbriga järgi kahanevalt.

Tabellite struktuur: Vaata andmeid ▲

töötajad
Töötajate (emp) ja nende osakondade (dept) andmebaas. Iga osakond on seotud null või rohkema töötajaga. Iga töötaja on seotud null või ühe osakonnaga.

Vihjed ▼
Lahendus ▼

Lahendus: Katseid kokku: 2
Lahenduse kood (ainult PostgreSQL): Lahendamiseks kulunud aeg: 0:13

1
2
3
4
5
6
7
8
9
10
11

Käivita lause

Joonis 11. Ülesande täitmise leht enne viimaste täienduste tegemist.

scott 003 x +

!52.127/practice/index.php/tasks/3?collection_id=12

Ülesanne: [Saada teade](#)
Ülesande kirjeldus: Vaata oodatud tulemust ▼
Tabellite struktuur ▼
Vihjed ▲
Vaata kõiki vihjeid Vaata vihjeid üksikvaal

Sorteerimise klausel on ORDER BY
Kahanevalt sorteerimiseks on DESC

Lahendus
Vaata kõiki lahendusi Vaata lahendusi üksikvaal

Korreleeruv alampäring.

```

SELECT
  deptno,
  dname
FROM
  Dept
WHERE
  NOT EXISTS (
    SELECT
      deptno
    FROM
      Emp
    WHERE
      Emp.deptno = Dept.deptno
  )
ORDER BY
  deptno DESC;

```

Lahendus: Katseid kokku: 2
Lahenduse kood (ainult PostgreSQL): Lahendamiseks kulunud aeg: 1:09

1
2
3
4
5
6
7
8
9
10
11

SELECT * FROM dept WHERE

dept
dept1
emp
ALTER
AND
AS
ASC
BETWEEN
BY
COUNT
CREATE
DELETE
DESC

Käivita lause

Küsimuste korral pöörduge aadressile Erki.Eessaar@taltech.ee.
© 2024 TalTech - Sellel lehel avaldatud materjali ilma loata kasutamine ja levitamine on keelatud.

Joonis 12. Ülesande edenemise leht avatud vihjete ja võimalike lahendustega.

SQLi Iseiseise Harjutamine Pealeht Ülesannete loend Ajalugu Statistika Programmist Logi välja

Ülesande nimi: scott 003 Lause tüüp: SELECT Andmebaas: töötajad Lahendamise seisund: Proovimata LAHENDAMISE AJALUGU

Ülesanne: Saada teade
 Ülesande kirjeldus: Vaata oodatud tulemust
 Leia kõik osakonnad, kus ei tööta ühtegi töötajat. Väljastage osakonna kood ja nimetus. Kasutage selleks EXISTS predikaati koos korreleeruva alampäringuga. Sorteerige tulemus osakonna numbrilise järjekorrale.
 Tabelite struktuur: Vaata andmeid
töötajad
 Töötajate (emp) ja nende osakondade (dept) andmebaas. Iga osakond on seotud null või rohkema töötajaga. Iga töötaja on seotud null või ühe osakonnaga.

 Vihjed: Vaata kõiki vihjeid Vaata vihjeid üksikvaas
 Valige, kuidas soovite vihjeid vaadata

Lahendus: Katselid kokku: 9
 Lahenduse kood (ainult PostgreSQL): Lahendamiseks kulunud aeg: 0:36

```
1 SELECT
```


Käivita lause

Tulemus:
 Staatus: ✗ Lahendamata Ridade arv:
 * Lause ei vasta ülesande tüübile:
 Lauseid ei saa käivitada!
 * Lause ei vasta kõigile ülesande-spetsiifilistele nõutud mustritele:
 Lauseid ei saa käivitada!
 Peab sisaldama EXISTS predikaati
 * Lause puudub nõutud sorteerimine:
 SELECT lause korral. Lauseid ei ole põhjust käivitada!
Võrdle oodatud tulemuselga

Koodi käivitamise tulemus:

Joonis 13. Ülesande täitmise leht, kui katse ebaõnnestub.

SQLi Iseiseise Harjutamine Pealeht Ülesannete loend Ajalugu Statistika Programmist Logi välja

Ülesande nimi: scott 046 Lause tüüp: UPDATE Keerukus: Edasijõudnutele Andmebaas: töötajad Lahendamise seisund: Lahendatud

Ajalugu: Vaata ülesannet
 Viimane lahendus: ✓ Lahendatud Katsed: 13 04.05.2024, 22:00:00
 Kulutatud aeg: 06:56 Võrdle õige vastusega

```
UPDATE
emp
SET
sal = sal * 1.10
FROM
dept
WHERE
emp.deptno = dept.deptno
AND dept.loc = 'DALLAS';
```

Selle ülesande ajalugu:

Katsed	Staatus	Lahendus
04.05.2024, 22:00:00	✓ Lahendatud	Rohkem detaile
04.05.2024, 21:59:53	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:59:20	✓ Lahendatud	Rohkem detaile
04.05.2024, 21:58:44	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:57:31	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:56:42	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:56:31	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:55:20	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:54:51	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:54:34	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:53:59	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:53:24	✗ Lahendamata	Rohkem detaile
04.05.2024, 21:52:45	✗ Lahendamata	Rohkem detaile

Joonis 14. Ülesande täitmise ajaloo leht.

Ajalugu

252.127/practice/index.php/attempts

SQLi Iseseisev Harjutamine

Algusaeg: 17.05.2024 00:00, Lõpu aeg: 19.05.2024 23:59, Lahendamise seisund: Pole valitud

OTSI

Tühjenda ajalugu

Ülesannete lahenuskatsed: 5

Kuupäev	Nimi	Lause tüüp	Keerukus	Andmebaas	Lahendamise seisund
19.05.2024, 12:58:57	scott 003	SELECT	Edasijõudnutele	töötajad	✓ Lahendatud
19.05.2024, 12:58:28	scott 003	SELECT	Edasijõudnutele	töötajad	✗ Lahendamata
19.05.2024, 12:58:24	scott 003	SELECT	Edasijõudnutele	töötajad	✓ Lahendatud
19.05.2024, 12:58:10	scott 003	SELECT	Edasijõudnutele	töötajad	✗ Lahendamata
18.05.2024, 11:57:41	scott 051	SELECT	Eksperttase	töötajad	✗ Lahendamata

Küsimuste korral pöörduge aadressile Erki.Fessaar@taltech.ee.
 © 2024 TalTech - Sellel lehel avaldatud materjali ilma loata kasutamine ja levitamine on keelatud.

Joonis 15. Kõigi ülesannete ajaloo leht.

Statistika

252.127/practice/index.php/statistics

SQLi Iseseisev Harjutamine

KOKKU KATSETUSED: 184, EDUKAD KATSETUSED: 20 (11%), EBANÕNNESTUNUD KATSETUSED: 164 (89%)

Kokku kulutatud aeg: 36:07:00, Õige lahenduseni jõudmise aeg: 19:41:00

Keerukus	Kõik ülesanded	Lõpetatud ülesanded	Protsent
Algfaase	25	2	8%
Edasijõudnutele	28	8	29%
Eksperttase	3	0	0%

SELECT 15% (Täidetud ülesanded: 6)

INSERT 0% (Täidetud ülesanded: 0)

UPDATE 67% (Täidetud ülesanded: 2)

DELETE 0% (Täidetud ülesanded: 0)

MERGE 100% (Täidetud ülesanded: 1)

CREATE TABLE 0% (Täidetud ülesanded: 0)

ALTER TABLE 25% (Täidetud ülesanded: 1)

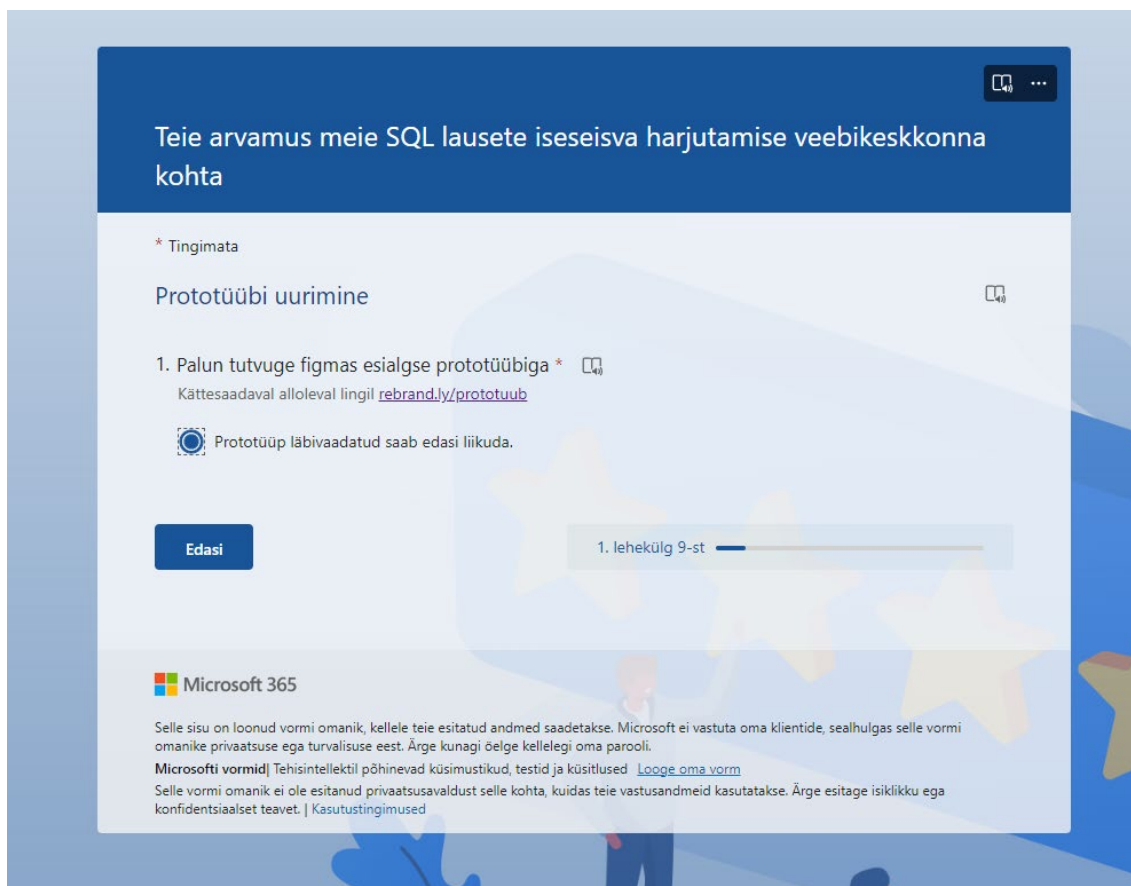
DROP TABLE 0% (Täidetud ülesanded: 0)

Andmebaasi nimi	Kõik ülesanded	Lõpetatud ülesanded	Protsent
töötajad	50	10	20%
koor	5	0	0%
hotell	1	0	0%

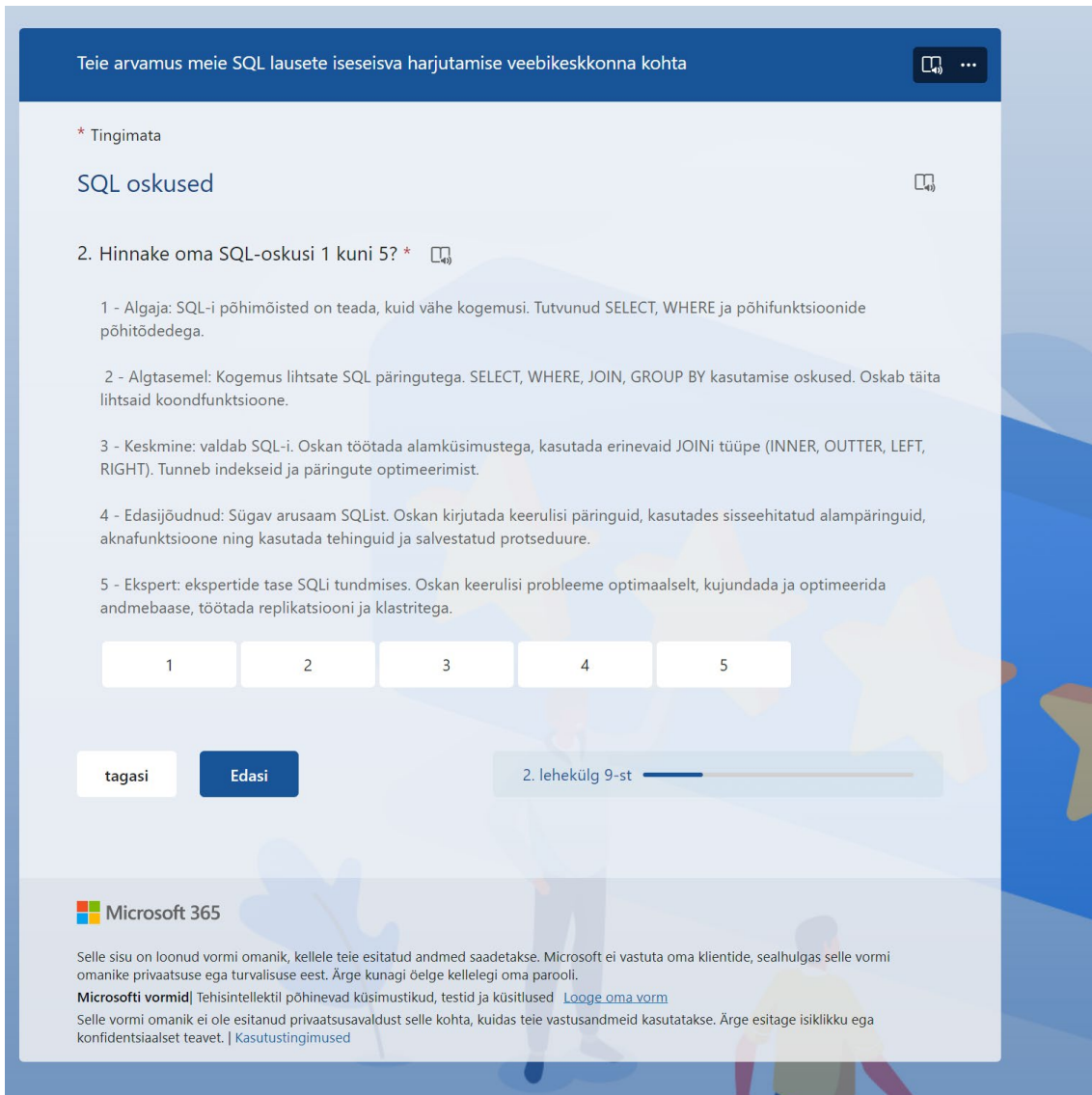
Kollektsiooni nimi	Kõik ülesanded	Lõpetatud ülesanded	Protsent
2024_harjutus_punktidele	7	7	100%
Aknafunktsioonid	2	1	50%
Alampäringud	11	1	9%

Joonis 16. Statistika leht.

Lisa 6 – Prototüübi küsimustik



Joonis 17. Prototüübi küsimustiku teine leht.



Joonis 18. Prototüübi küsimustiku kolmas leht.



* Tingimata

Pealeht



Palun kirjeldage oma muljeid selle lehel

3. Kuidas hindate teie sellel lehel? *

Hindamiskriteeriumid:

1 - Ei meeldinud üldse

2 - Ei meeldinud

3 - ei

meeldinud 4 - meeldis

5 - Väga meeldis

	1	2	3	4	5
Disain (värvide valik, font, kirjad)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Elementide interaktiivsus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Intuiitivne navigeerimine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lehekülje struktuur	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Esitatud teabe selgus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visuaalne atraktiivsus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kasutamise lihtsus (näiteks lihtne juurdepääs põhiosadele/toimingutele)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pealkirjade ja põhielementide atraktiivsus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Esitatud andmete selgus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. Mida võiksite sellel lehel parandada? Palun märkige konkreetsed aspektid või funktsioonid, mis teie arvates vajavad parandamist.

Sisesta oma vastus

tagasi

Edasi

Lk 3/9

Microsoft 365

Selle sisu on loonud vormi omanik, kellele teie esitatud andmed saadetakse. Microsoft ei vastuta oma klientide, sealhulgas selle vormi omanike privaatsuse ega turvalisuse eest. Ärge kunagi öelge kellelegi oma parooli.

Microsofti vormid | Tehisintellektil põhinevad küsimustikud, testid ja küsitlused [Looge oma vorm](#)

Selle vormi omanik ei ole esitanud privaatsusavaldust selle kohta, kuidas teie vastusandmeid kasutatakse. Ärge esitage isiklikku ega konfidentsiaalset teavet. | [Kasutustingimused](#)

Joonis 19. Prototüübi küsimustiku neljas leht.

Teie arvamus meie SQL lausetes iseseisva harjutamise veebikeskkonna kohta

* Tingimata

Üldine tagasiside

15. Kas on mõni funktsioon või funktsionaalsus, mis teie arvates meie platvormil puudub? (Palun täpsustage konkreetsed aspektid või funktsioonid): *

Jah

Ei

16. Kas te kasutaksite meie platvormi tulevikus, näiteks "Andmebaasi I" kursuse raames? *

Jah, sest see platvorm annab võimaluse parandada sql keeleoskust, rohkem praktikat

Ei

17. Kas te soovitaksite meie platvormi kolleegidele või sõpradele? *

Jah

Ei

Ei ole kindel

18. Teie kommentaar või ettepanek meie platvormi kohta (vabatahtlik, kuid teie arvamus on meile oluline)

Sisesta oma vastus

Pärast saatmist saate vastusest koopia välja printida

tagasi Saada Lk: 9/9

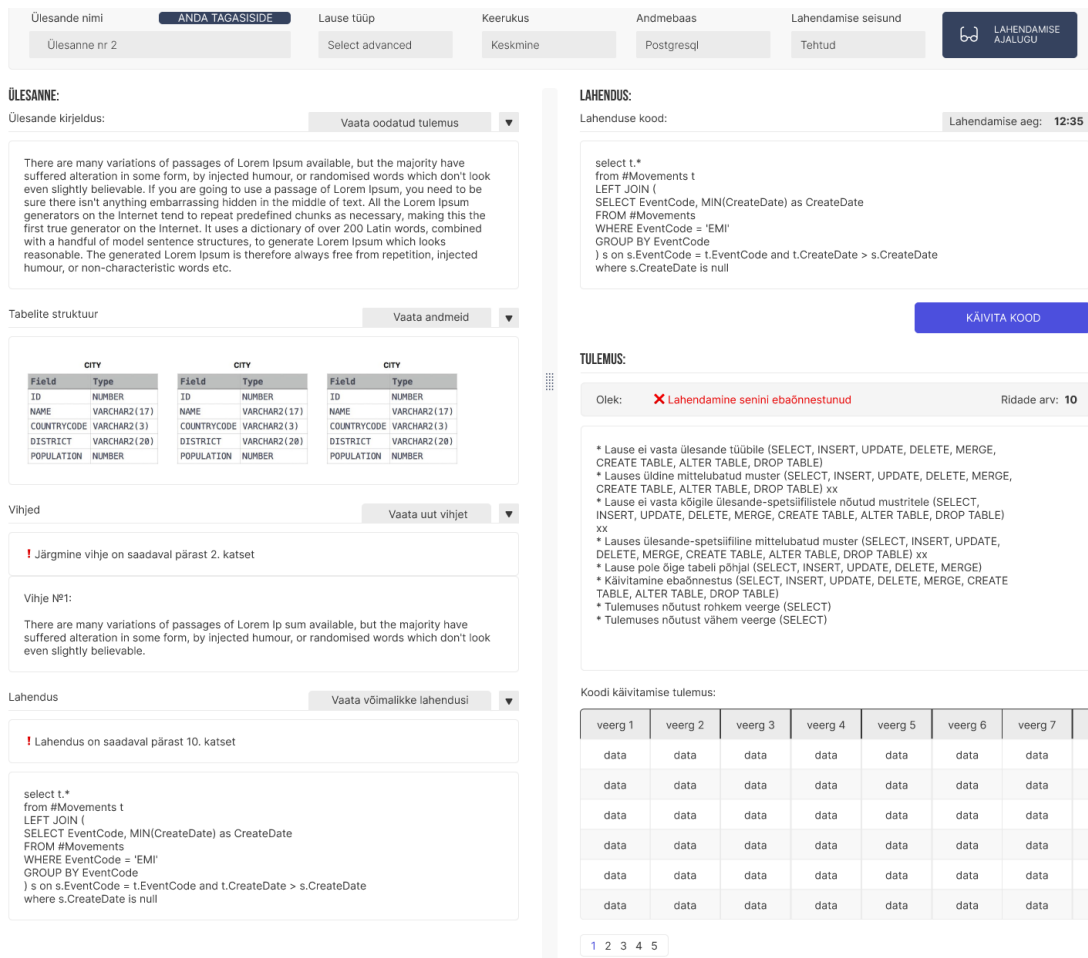
Microsoft 365

Selle sisu on loonud vormi omanik, kellele teie esitatud andmed saadetakse. Microsoft ei vastuta oma klientide, sealhulgas selle vormi omanike privaatsuse ega turvalisuse eest. Ärge kunagi öelge kellelegi oma parooli.

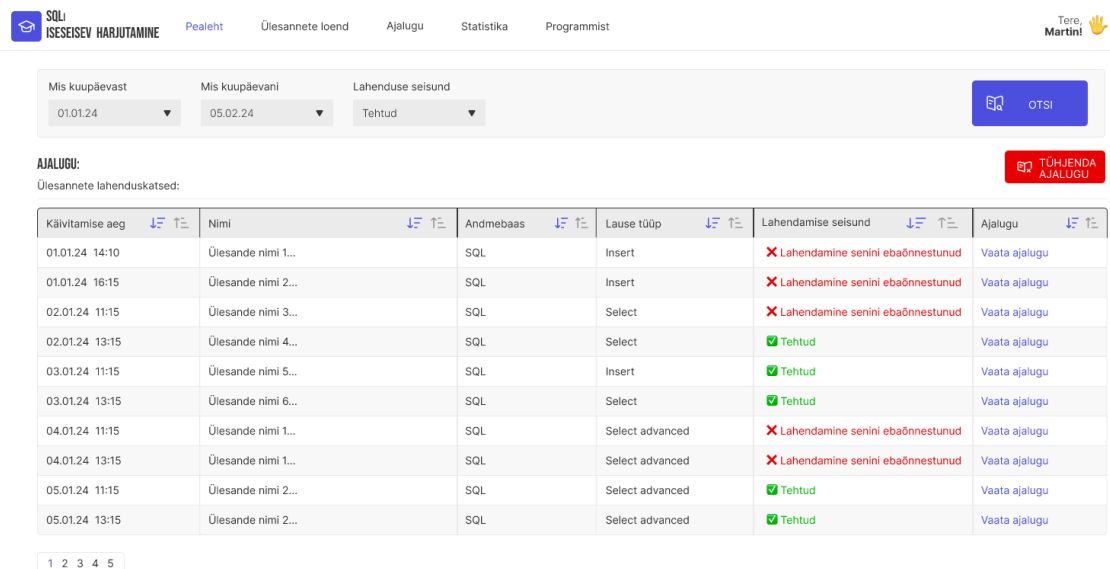
Microsofti vormid| Tehisintellektil põhinevad küsimustikud, testid ja küsitlused [Looge oma vorm](#)

Selle vormi omanik ei ole esitanud privaatsusavaldust selle kohta, kuidas teie vastusandmeid kasutatakse. Ärge esitage isiklikku ega konfidentsiaalset teavet. | [Kasutustingimused](#)

Joonis 20. Prototüübi küsimustiku viimane leht.



Joonis 23. Ülesande lehe prototüüp Figmas.



Joonis 24. Ajaloo lehe prototüüp Figmas.

Ülesande nimi: Ülesanne nr 2 | Lause tüüp: Select advanced | Keerukus: Keskmine | Andmebaas: Postgresql | Lahendamise seisund: Tehtud

ÜLESANDE AJALUGU: VAATA ÜLESANNET

Viimane lahendus: ✗ Lahendamine senini ebaõnnestunud | Katsed: 10 | 01.01.24 - 14:21:02

```
select t.*
from #Movements t
LEFT JOIN (
SELECT EventCode, MIN(CreateDate) as CreateDate
FROM #Movements
WHERE EventCode = 'EMI'
GROUP BY EventCode
) s on s.EventCode = t.EventCode and t.CreateDate > s.CreateDate
where s.CreateDate is null
```

Tagasiside:

- * Lause ei vasta ülesande tüübile (SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE, DROP TABLE)
- * Lauseis üldine mittelubatud muster (SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE, DROP TABLE) xx
- * Lause ei vasta kõigile ülesande-spetsiifilistele nõutud mustritele (SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE, DROP TABLE) xx
- * Lauseis ülesande-spetsiifiline mittelubatud muster (SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE, DROP TABLE) xx
- * Lause pole õige tabeli põhjal (SELECT, INSERT, UPDATE, DELETE, MERGE)
- * Käivitamine ebaõnnestus (SELECT, INSERT, UPDATE, DELETE, MERGE, CREATE TABLE, ALTER TABLE, DROP TABLE)
- * Tulemuses nõustus rohkem veerge (SELECT)
- * Tulemuses nõustus vähem veerge (SELECT)

Seile ülesande ajalugu:

Käivitamise aeg	Lahendamise seisund	Lahendus
01.01.24 - 14:21:02	✗ Lahendamine senini ebaõnnestunud	Rohkem detaile
01.01.24 - 14:21:02	✗ Lahendamine senini ebaõnnestunud	Rohkem detaile
01.01.24 - 14:21:02	✗ Lahendamine senini ebaõnnestunud	Rohkem detaile
01.01.24 - 14:21:02	✔ Tehtud	Rohkem detaile
01.01.24 - 14:21:02	✔ Tehtud	Rohkem detaile
01.01.24 - 14:21:02	✔ Tehtud	Rohkem detaile
01.01.24 - 14:21:02	✗ Lahendamine senini ebaõnnestunud	Rohkem detaile
01.01.24 - 14:21:02	✗ Lahendamine senini ebaõnnestunud	Rohkem detaile
01.01.24 - 14:21:02	✔ Ülesanne on täidetud	Rohkem detaile
01.01.24 - 14:21:02	✔ Ülesanne on täidetud	Rohkem detaile

1 2 3 4 5

Joonis 25. Ülesande ajaloo lehe prototüüp Figma.

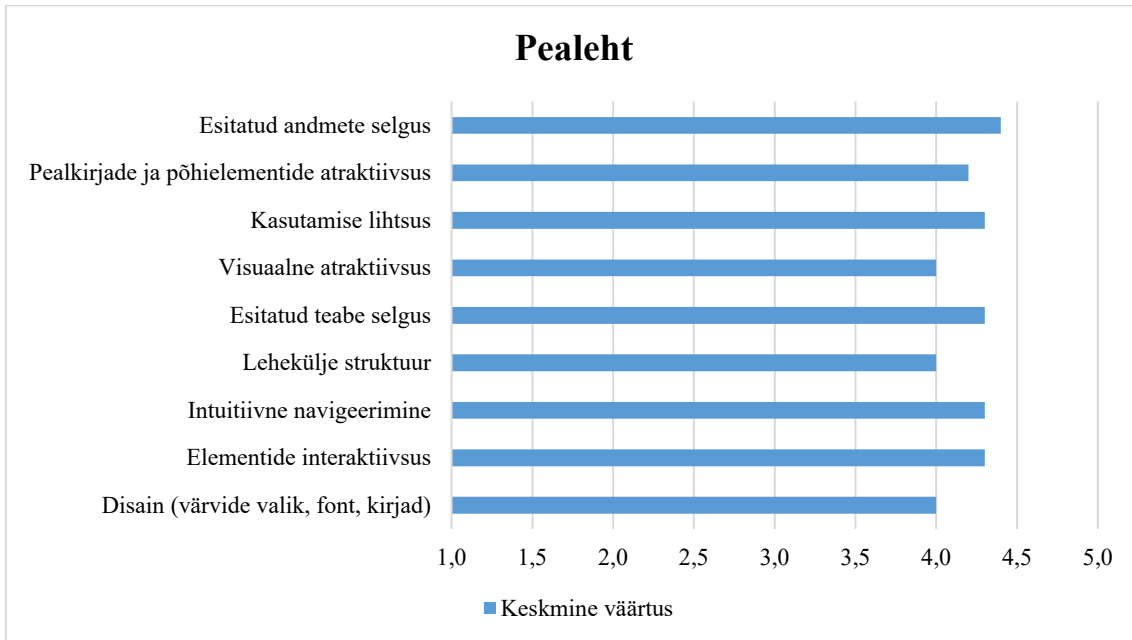
KOKKU KATSETUSED **1345** | EDUKAD KATSETUSED **890 - 78%** | EBAÕNNESTUSED KATSETUSED **340 - 22%**

- PARIM TÄITMISAE:**
- 13:32 - parim aeg **kergete** ülesannete täitmiseks
 - 16:32 - parim aeg **keskmiste** ülesannete täitmiseks
 - 19:32 - parim aeg **keerukate** ülesannete täitmiseks

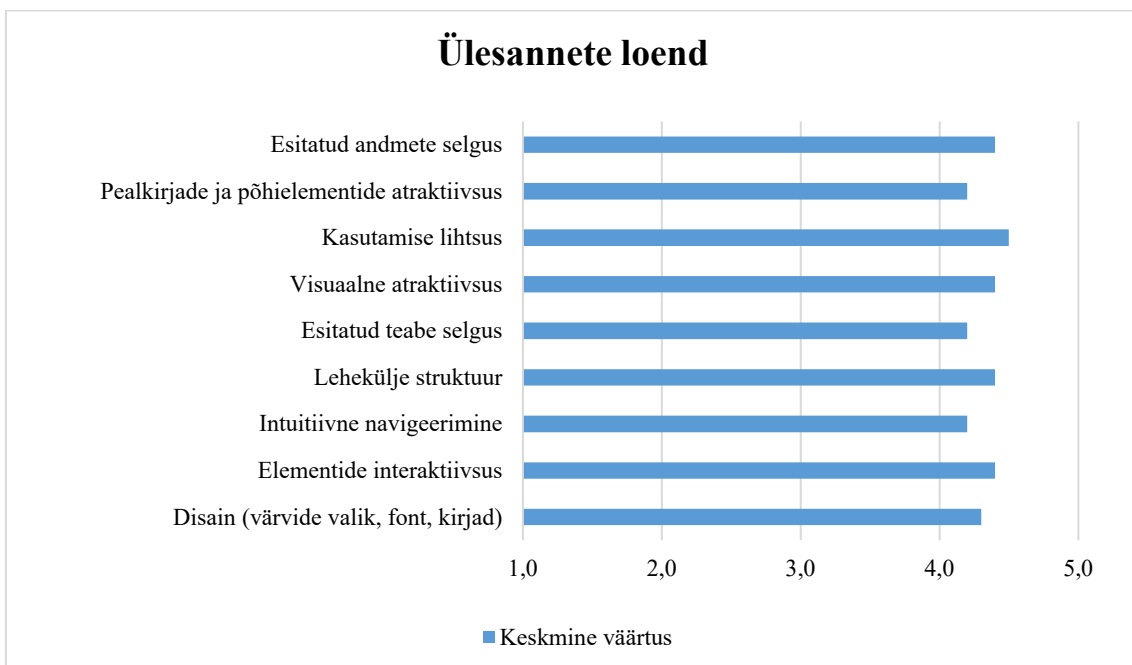
- KESKMINE VALMIMISAE:**
- 13:32 - parim aeg **kergete** ülesannete täitmiseks
 - 16:32 - parim aeg **keskmiste** ülesannete täitmiseks
 - 19:32 - parim aeg **keerukate** ülesannete täitmiseks

Joonis 26. Statistika lehe prototüüp Figma.

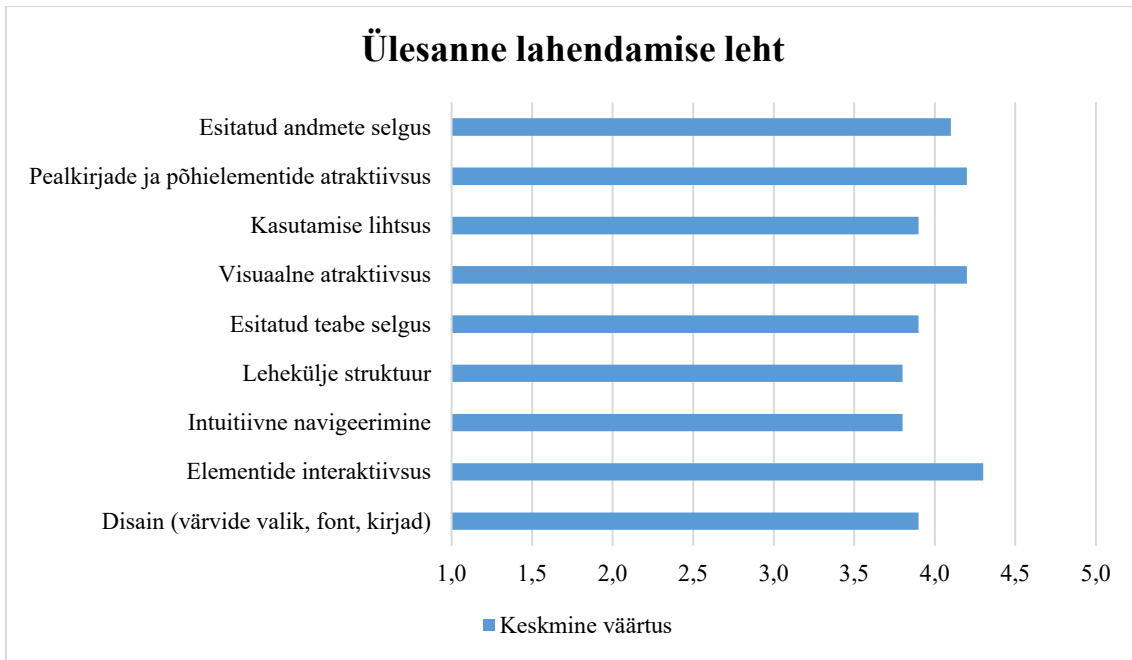
Lisa 8 – Prototüübi uuringu tulemustel põhinevad tulpdiaграмmid



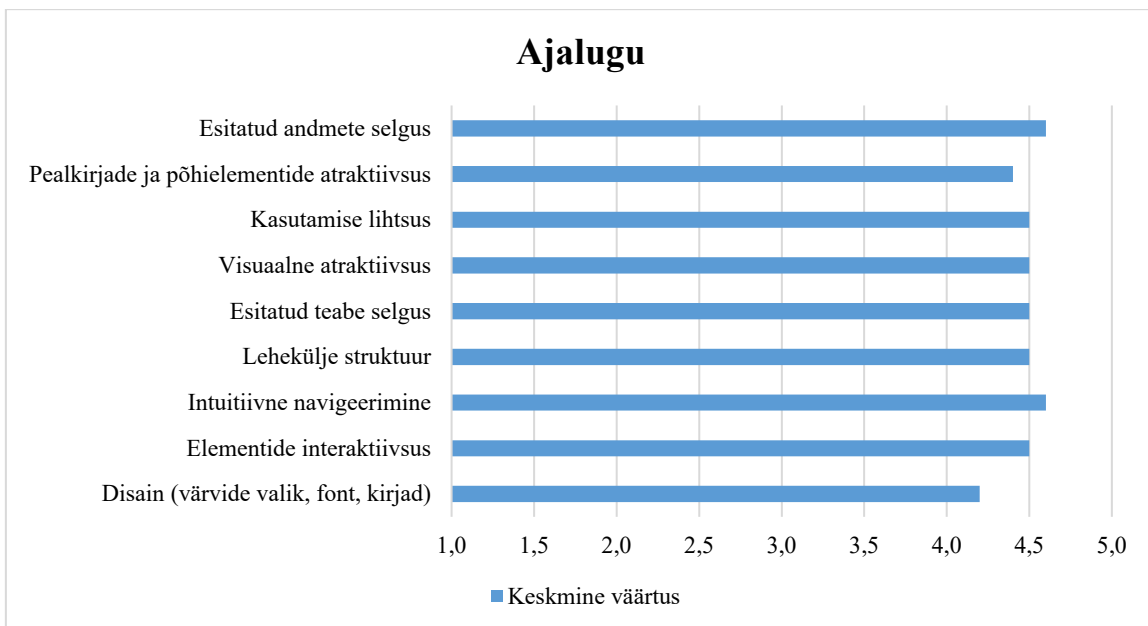
Joonis 27. Hinnangud pealehe kohta.



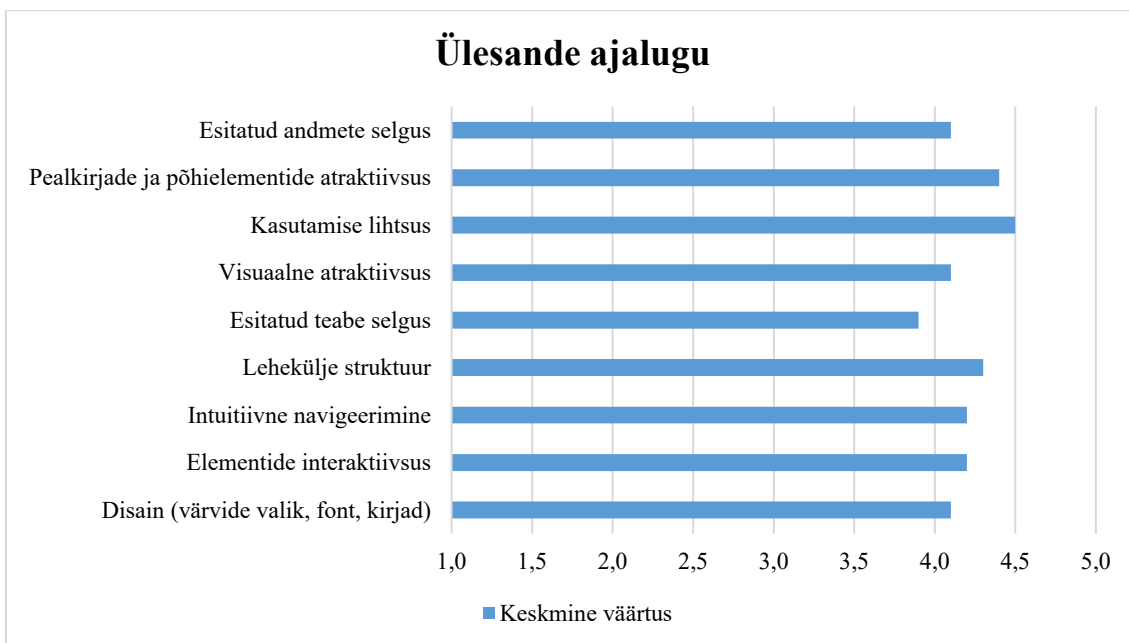
Joonis 28. Hinnangud ülesannete loendi kohta.



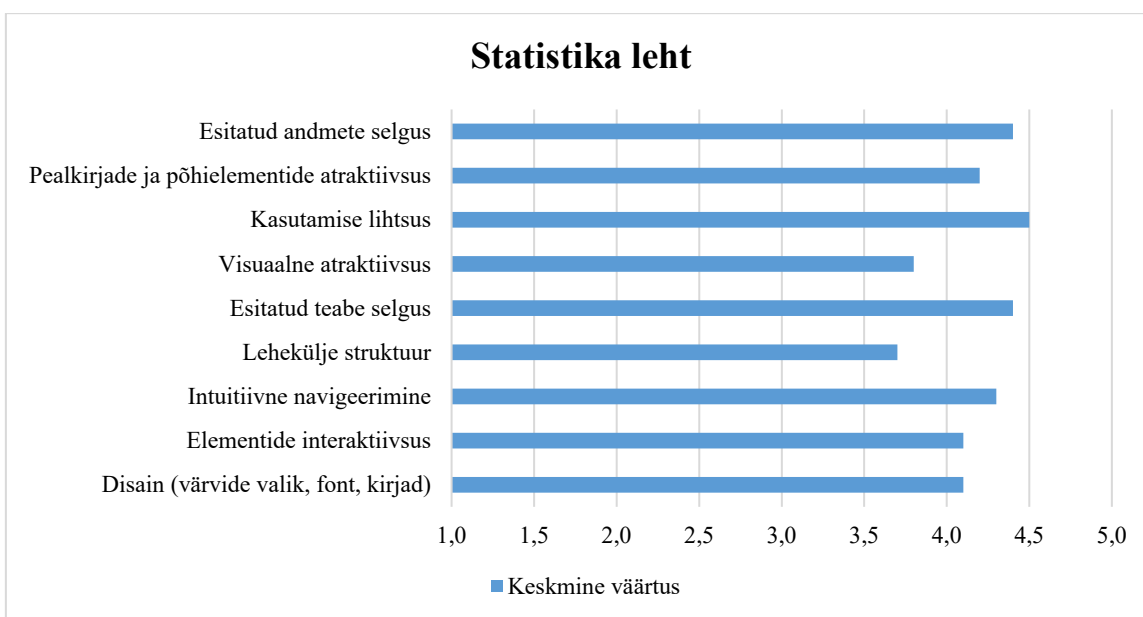
Joonis 29. Hinnangud ülesanne lahendamise lehe kohta.



Joonis 30. Hinnangud ajaloo lehe kohta.



Joonis 31. Hinnangud ülesande ajaloo lehe kohta.

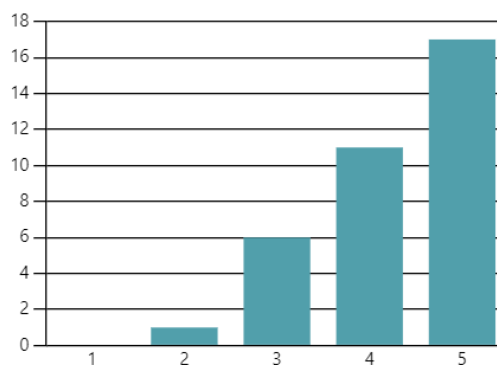


Joonis 32. Hinnangud statistika lehe kohta.

Lisa 9 – Veebirakenduse uuringu tulemustel põhinevad tulpdiagrammid

2. Millise mulje jättis Teile "Pealeht"?

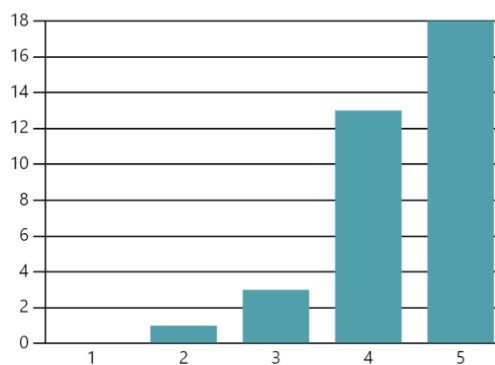
4.26
Keskmine hinne



Joonis 33. Hinnangud pealehe kohta.

4. Millise mulje jättis Teile ülesannete loendi leht?

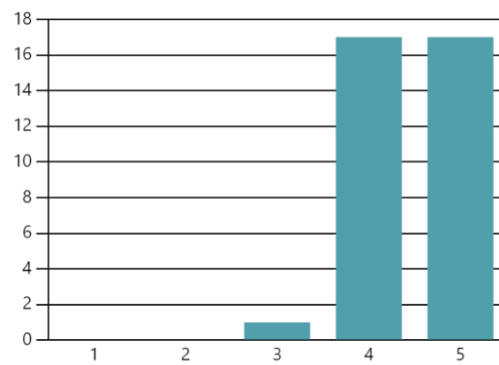
4.37
Keskmine hinne



Joonis 34. Hinnangud ülesannete loendi lehe kohta.

6. Millise mulje jättis Teile ülesande lahendamise leht?

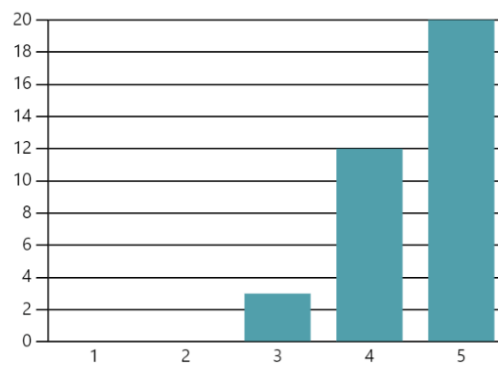
4.46
Keskmine hinne



Joonis 35. Hinnangud ülesande lahendamise lehe kohta.

8. Millise mulje jättis Teile ülesannete ajaloo leht?

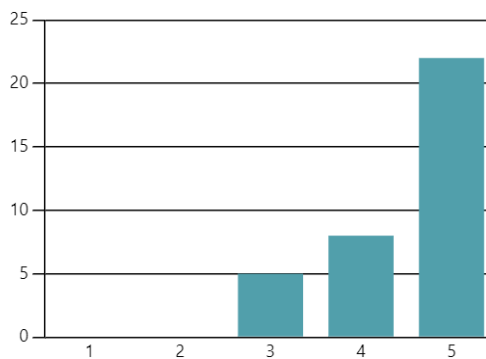
4.49
Keskmine hinne



Joonis 36. Hinnangud ülesannete ajaloo lehe kohta.

10. Millise mulje jättis Teile ülesande ajaloo leht?

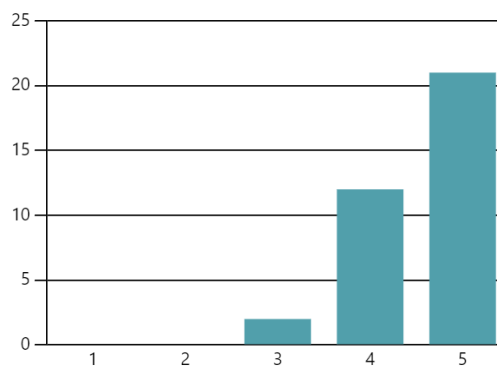
4.49
Keskmine hinne



Joonis 37. Hinnangud ülesande ajaloo lehe kohta.

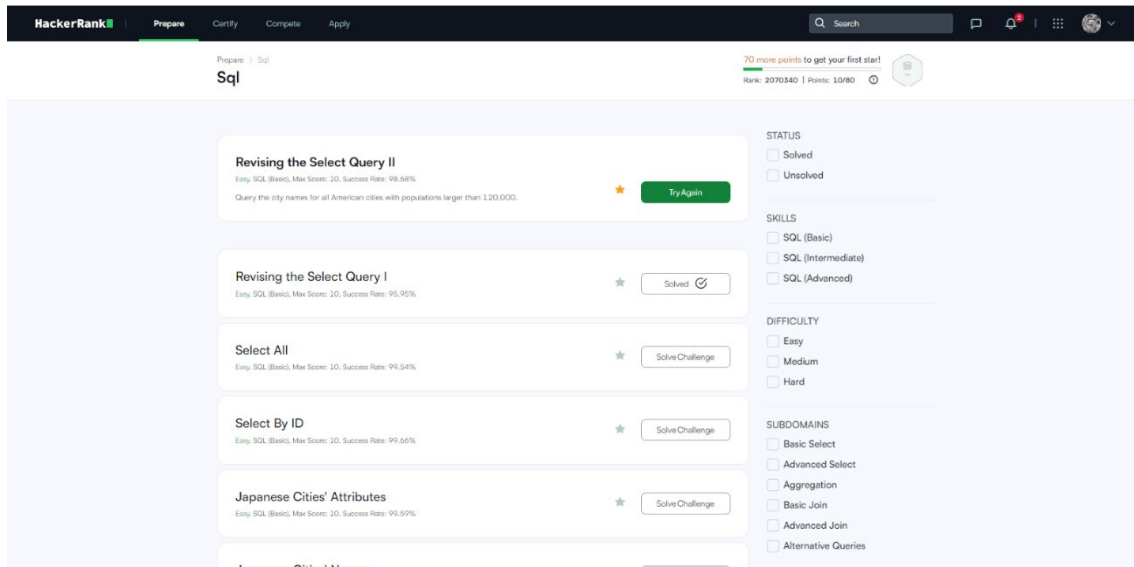
12. Millise mulje jättis Teile statistika leht?

4.54
Keskmine hinne

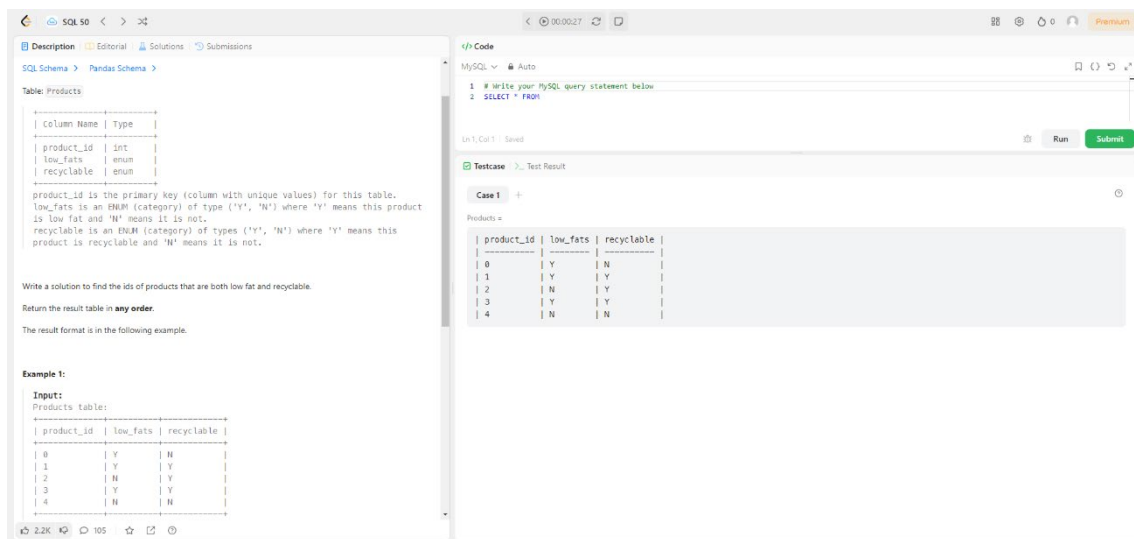


Joonis 38. Hinnangud statistika lehe kohta.

Lisa 10 – Olemasolevate SQLi harjutamiste rakenduste kasutajaliides



Joonis 39. HackerRank ülesannete loendi leht.



Joonis 40. LeetCode platvormi ülesande lahenduse leht, kus tabelid on tekstivormis.

Retrieve everything from a table ✔

Question

How can you retrieve all the information from the cd.facilities table?

cd.members

memberid	integer
surname	character varying(255)
firstname	character varying(255)
address	character varying(200)
zipcode	integer
telephone	character varying(20)
recommendedby	integer
joindate	timestamp

cd.bookings

facid	integer
memberid	integer
starttime	timestamp
slotid	integer

cd.facilities

facid	integer
name	character varying(100)
membercost	numeric
guestcost	numeric
initialoutlay	numeric
monthlymaintenance	numeric

Expected Results

facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
0	Tennis Court 1	5	25	10000	200
1	Tennis Court 2	5	25	8000	200
2	Badminton Court	0	15.5	4000	50
3	Table Tennis	0	5	320	10
4	Massage Room 1	35	80	4000	3000
5	Massage Room 2	35	80	4000	3000
6	Squash Court	3.5	17.5	5000	80
7	Snooker Table	0	5	450	15
8	Pool Table	0	5	400	15

Your Answer 🔍 📄 📌 🏃

```
select * from cd.facilities;
```

Answers and Discussion 🔍

```
select * from cd.facilities;
```

The `SELECT` statement is the basic starting block for queries that read information out of the database. A minimal select statement is generally comprised of `select` (some set of columns) from (some table or group of tables).

In this case, we want all of the information from the facilities table. The `FROM` section is easy - we just need to specify the `cd.facilities` table. `cd` is the table's schema - a term used for a logical grouping of related information in the database.

Next, we need to specify that we want all the columns. Conveniently, there's a shorthand for 'all columns' - `*`. We can use this instead of laboriously specifying all the column names.

Joonis 41. PostgreSQL Exercises ülesande täitmise leht.

SQLPad Questions ▾ AI Tools ▾ Blog Courses Pricing 🔥
Light Dark
UPGRADE
Hello, leonid0 230 ▾

walmart

Single Table: SELECT, WHERE, AND, OR, ORDER BY, LIMIT ✕

Show 100 entries Search:

ID	Title	Level	FTPR	Lecture	Solved
1	Top store for movie sales	easy	41%	2,1	✕
2	Top 3 movie categories by sales	easy	42%	2,1	✕
3	Top 5 shortest movies	easy	49%	2,1	✕
4	Staff without a profile image 🔒	easy	66%	2,1	✕
16	Staff who live in Woodridge	easy	65%	2,1	✕
17	GROUCHO WILLIAMS' actor_id	easy	57%	2,1	✕
92	Acceptance rate first week of 2021 🔒 spotify	easy	14%	2,1	✕
111	Second most popular product 🔒 amazon	medium	16%	2,1	✕
126	Top 3 restaurants 🔒 uber	hard	10%	2,1	✕
152	Salary analysis 🔒 dropbox	easy	30%	2,1	✕

Joonis 42. SqlPad ülesannete loendi leht.

Lisa 11 – Olemasolevate vahendite võrdlustabel

Tabel 5. Olemasolevate vahendite võrdlustabel.

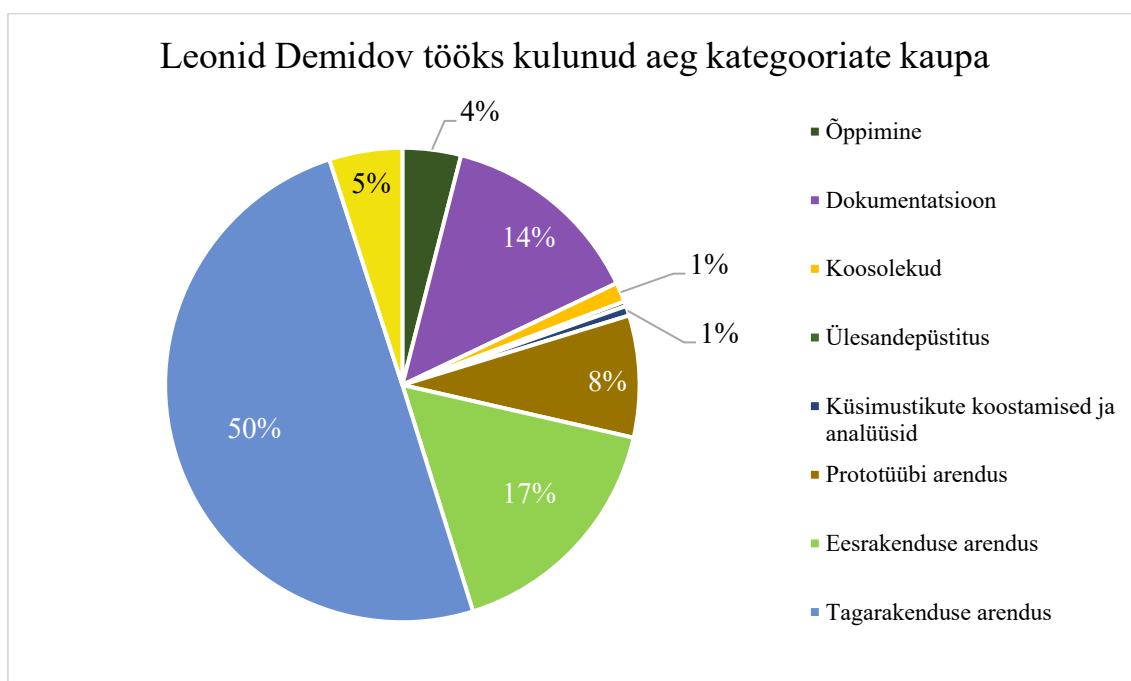
Süsteem	Loodud	HackerRank	LeetCode	PostgreSQL Exercises	SQLPad	TÜ bakatöö
Toetatud lausete tüübid	SELECT INSERT UPDATE DELETE MERGE CREATE TABLE ALTER TABLE DROP TABLE	SELECT	SELECT	SELECT	SELECT	SELECT INSERT UPDATE DELETE CREATE TABLE
Andmebaasisüsteemid	PostgreSQL	MySQL DB2 Oracle MS SQL Server	MySQL	PostgreSQL	PostgreSQL MySQL	PostgreSQL
Tasuta ülesannete arv	piiramatu	57	50	60	230	13
Ülesannete lisamise võimalus õppejõu poolt	Jah	Ei	Ei	Ei (eeldab programmeerimist)	Ei	Ei (eeldab programmeerimist)
Ülesannete otsimise filtrid	Nimi Andmebaas Lause tüüp Raskusaste Lahendamise seisund Kogum	Täitmise olek Keerukus Oskuse tase Select tüüp	Keerukus nimi	puudub	Tekstiotsing ettevõtte põhjal Lahendamise seisund (oma konto alt)	Ei
Enda lahendamise ajalugu	Jah	Jah	Jah	Ei	Jah	Ei

Süsteem	Loodud	HackerRank	LeetCode	PostgreSQL Exercises	SQLPad	TÜ bakatöö
Enda lahendamiste statistika	Jah	Jah	Jah	Ei	Jah	Ei
Mitmekeelsus (ülesanded ja kasutajaliides)	Jah	Ei	Ei	Ei	Ei	Ei
Hinne süsteemi poolt lahendusele antava tagasiside sisukusele (5 palli skaalal)	4	3	3	4	3	3
Vihjete vaatamine	Jah	Ei	Jah	Jah	Ei	Ei
Võimalike lahenduste vaatamine	Jah	Ei	Jah	Jah	Jah (tasu eest)	Ei

Lisa 12 – Leonid Demidovi panuse kirjeldus ja eneseanalüüs

Lõputöö projektis ei olnud täpset ülesannete jaotust. Iga meeskonnaliige osales erinevates töötappides, suured ülesanded jaotati väiksemateks osadeks ja jagati projektis osalejate vahel. Selline lähenemine võimaldas igal osalejal saada tervikliku ülevaate veebirakenduse loomisprotsessist.

Projekti elluviimiseks kulus mul kokku neli kuud. Joonisel 43 on täpsemalt näidatud aja jaotus protsentuaalselt erinevate projekti kategooriate vahel. Kõige rohkem aega kulus serveripoolse arenduse peale (50%), kasutajaliidese loomisele (17%) ja dokumentatsiooni koostamisele (14%).



Joonis 43. Leonid Demidovi tööks kulunud aja jaotus.

Varasem kogemus PHP projektidega võimaldas mul kiiresti meenutada keele põhimõtteid. Alustasin olemasolevate SQLi harjutamiseks mõeldud platvormide otsimise ja analüüsiga. Pärast platvormide analüüsi alustasime prototüübi loomist Figma's, millega tegelesin peamiselt mina tänu oma kogemusele kursusel „Start-up ettevõtlus TMJ0190“.

Paralleelselt tudengite küsitluse läbiviimisega Figma prototüübi analüüsiks alustasin projekti käivitamist: failisüsteemi ja keskkonna seadistamine ning Docker'i

integreerimine. Valisin projekti arhitektuuriks MVC lähenemise REST API kasutamisega kliendi ja serveri vaheliseks suhtluseks, tuginedes kursusel “Infosüsteemide arendamine” saadud kogemusele.

Arendus algas serveripoolse osa ja üldiste koodimoodulite loomisega, mille eest vastutasin mina. Lõin kasutaja sessiooni seadistamise, mitmekeelsuse ning serveri kontrollerite ja mudelite vanemklassid, mis hõlmasid kliendi vigade töötlemist, andmebaasiga ühenduse loomist ja mugavat andmebaasiga suhtlemist. Lõpp-punktide loomise ülesanded jagunesid võrdselt, mina vastutasin ülesannete loendi ja täitmise lehtede eest, kasutades kursusel „Andmebaasid I” saadud teadmisi.

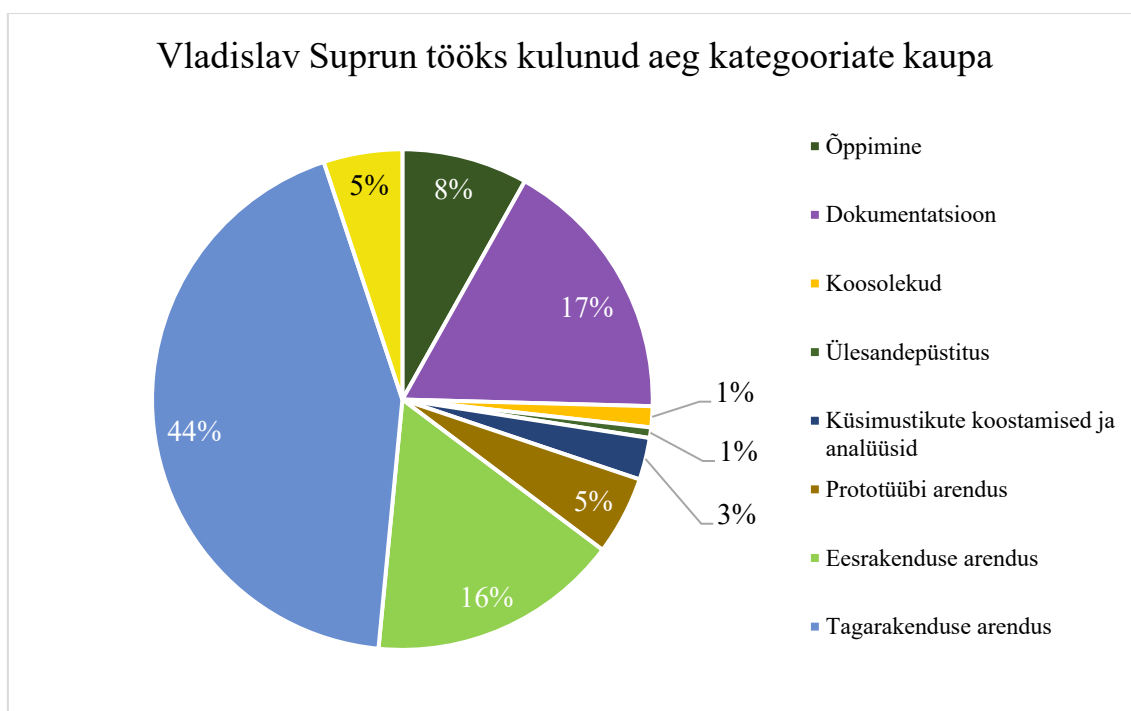
Pärast lõpp-punktide loomist kirjutasin Postman'is testid serveri vastuste testimiseks. Ülejäänud lõpp-punkte testis minu kaaslane. Pärast edukat testide läbimist alustasime kasutajaliidese loomist Figma prototüübi põhjal ja lõpp-punktide ühendamist. Minu ülesanded hõlmasid ülesande täitmise, ülesannete loendi ja üksiku ülesande täitmise ajaloo lehti. Lõpetuseks viisime töötava kliendi ja serveriga veebisaidi demoversiooni minu isiklikule hostimisele, et süsteemi kõigi meeskonnaliikmete poolt mugavalt testida. Dokumendi kallal tööga tegelesime ühiselt.

See projekt oli suurepärane võimalus rakendada ülikoolis omandatud teadmisi praktikas ja süvendada oma teadmisi veebirakenduste arenduses.

Lisa 13 – Vladislav Suprun panuse kirjeldus ja eneseanalüüs

Lõputöö koostamisel ei olnud täpsed rollid määratud. Paljusid ülesandeid täideti ühiselt või jaotati suurema ülesande sees väiksemateks tegevusteks. See andis võimaluse osaleda projekti kõikides etappides ja õppida palju uut ning huvitavat.

Projekti valmimiseks kulus mul kokku neli kuud. Joonis 44 näitab täpsemalt aja jaotust protsentuaalselt erinevate kategooriate vahel. Kõige rohkem aega kulus tagarakenduse arendamisele, dokumentatsioonile ja kasutajaliidese loomisele.



Joonis 44. Vladislav Supruni tööks kulunud aja jaotus.

Minu teekond projekti raames algas PHP õppimisega. Väga kasulikuks osutus varasem kogemus C# ja Java keeltega, kuid siiski õppisin palju uut. Alguses alustasime olemasolevate SQL-treenimiseks mõeldud veebisaitide uurimise, testimise ja analüüsimisega, jagades ülesandeid võrdselt paariliste vahel.

Pärast prototüübi loomist oli minu ülesandeks küsitleda tulevasi kasutajaid ning teha järeldusi, mida saaks meie lahenduses täiustada ja lisada. Veebisaidi arendamine jaotati ülesanneteks ja alamülesanneteks. Iga etapi lõpus kogunesime Teamsi keskkonda, kus testisime ja ühendasime erinevad arendusoskjad. Esimese projekti versiooni valmimisel viisin läbi tudengite küsitluse ning tegin selle põhjal järeldused. Samuti lõin koos

paarilisega uusi ülesandeid funktsionaalsuse parandamiseks ja täiustamiseks ning analüüsisin tarkvara edasise arendamise võimalusi.

Kokkuvõttes oli töö SQL-iseseisva praktika platvormi loomise projekti kallal väga mahukas, viljakas ja huvitav, pakkudes uusi teadmisi ja kogemusi. Eriti meeldis koostöö paaris, kuna see võimaldas mitte ainult juhendajaga konsulteerida, vaid ka kaasautoriga erinevaid küsimusi arutada. Hindamatuks kogemuseks oli teadmiste ja oskuste vahetamine ning vastastikune abi keerulistes olukordades. Projekti lõpuleviimine tõi kaasa väärtuslikud oskused ja teadmised, mis on olulised edasiseks professionaalseks arenguks IT valdkonnas.