

Tartu Ülikool
Arvutiteaduse instituut
Informaatika õppekava

Johan Hollak

Ainesse “Andmebaasid” näidisrakenduse
loomine male andmebaasi põhjal

Bakalaureusetöö (9 EAP)

Juhendaja:
Piret Luik, PhD

Tartu 2025

Ainese “Andmebaasid” näidisrakenduse loomine male andmebaasi põhjal

Lühikokkuvõte:

Bakalaureusetöö eesmärk oli luua Tartu Ülikooli kursuse “Andmebaasid” jaoks veebirakendus, mida saaks kasutada üliõpilastele andmebaaside olulisuse demonstreerimiseks ühe rakenduse toimimisel. Samuti võeti eesmärgiks koostada õppijatele ülesanne rakendust laiendada. Töö raames arendati vastav rakendus ning see võeti juba kasutusele 2025 kevadsemestril Tartu Ülikooli aines "Andmebaasid". Tagasiside põhjal selgus, et kursuse õppejõudude arvates oli rakendus hästi tehtud, loogiliselt üles ehitatud ning oma eesmärki täitev ning ülesanne selgelt koostatud ning ka üliõpilaste tagasiside oli pigem positiivne.

Võtmesõnad: andmebaas, veebirakendus, õppevahend

CERCS: P175 Informaatika, süsteemiteooria; S270 Pedagoogika ja didaktika

Developing a web application for use in Databases course based on the sample chess database

Abstract:

The aim of the Bachelor’s thesis was to create a web application for the University of Tartu course “Databases” that could be used to demonstrate to students the importance of databases in the operation of an application. Another goal was to prepare an assignment for learners to extend the application. As part of the thesis, the application was developed and was already introduced in the spring semester of 2025 in the University of Tartu course “Databases”. Feedback showed that the course instructors found the application well made, logically structured, and fit for its purpose, and the assignment clearly formulated; students’ feedback was also generally positive.

Keywords: database, web application, study material

CERCS: P175 Informatics, systems theory; S270 Pedagogy and didactics

Sissejuhatus	4
Mõisted ja terminid	5
1. Teoreetiline ülevaade	6
1.1 Rakenduse olulisus lisaks andmebaasile	6
1.2 Rakenduse arhitektuuri mõiste	8
1.3 Mitmekihilise rakenduse arhitektuuri põhimõtted	8
1.4 Andmebaas süsteemi osana	10
1.5 Andmebaaside õpetamine Tartu Ülikoolis	11
2. Rakenduse loomine	13
2.1 Nõuded rakendusele	13
2.1.1 Funktsionaalsed nõuded	13
2.1.2 Mittefunktsionaalsed nõuded	14
2.2 Rakenduse ülesehitus	14
2.3 Valminud rakenduse arhitektuur	17
2.4 Arendamise protsess	19
3. Valminud lahendus	21
3.1 Maleturniiride rakendus	21
3.2 Boonusülesanne üliõpilastele	23
3.3 Rakenduse kasutamine õppevahendina	24
3.4 Tagasiside rakendusele	25
3.4.1 Küsimustik praktikumijuhendajatele	25
3.4.2 Küsimustik üliõpilastele	29
Kokkuvõte	31
Viidatud kirjandus	32
Lisad	34
1. Rakenduse lähtekood	34
2. Küsimustik üliõpilastele	34
3. Küsimustik praktikumijuhendajatele	34
Litsents	35

Sissejuhatus

Andmebaasidel on vaieldamatult oluline roll tänapäevase maailma toimimises, ning andmebaaside õppimine on iga ülikooli arvutiteaduse bakalaureuseõppe osa. Tartu Ülikoolis kuulub kursus “Andmebaasid (LTAT.03.004)” [1] esimese kursuse informaatika bakalaureuseõppe üliõpilaste kohustuslike õppeainete hulka, ning seda läbitakse tavaliselt esimese õppeaasta kevadsemestril.

Andmebaaside õpetamisel alustatakse sellest, kuidas andmebaase luua ja kuidas andmeid nende sees muuta, kuidas andmebaas rakenduse osana toimib, jääb algkursuse raames tihti üksnes teooria tasemele. See tingib olukorra, kus üliõpilased ei saa aru, kuhu nende loodavad andmebaasid suuremas pildis paigutuvad ning see raskendab ka andmebaasi funktsionaalsustest arusaamist [2, 3]. Sellise probleemi olemasolu on kirjeldanud ka Tartu Ülikooli kursuse “Andmebaasid” vastutav õppejõud professor Piret Luik, kes on ka käesoleva bakalaureusetöö juhendaja.

Bakalaureusetöö eesmärk on luua näitlikustav veebirakendus, mida saaks kasutada kursuse “Andmebaasid” käigus üliõpilastele selgema konteksti andmiseks, kuidas andmebaase tarkvaraarenduse maailmas kasutatakse. Rakendusega käib kaasas ka boonusülesanne, kus tuleb rakendust laiendada, andes õppuritele võimaluse oma teadmisi praktilise ülesandega täiustada.

Bakalaureusetöö esimeses peatükis kirjeldatakse praktiliste näidete ja selgema üldpildi loomise olulisust andmebaaside õpetamisel. Seejuures keskendutakse just andmebaaside rollile ühe rakenduse osana. Samas peatükis kirjeldatakse ka sellise rakenduse loomise olulisi aspekte ning tutvustatakse lühidalt andmebaaside õpetamist Tartu Ülikoolis. Bakalaureusetöö teises peatükis tutvustatakse töö käigus valminud rakenduse arendamisel tehtud valikuid ning rakenduse arhitektuuri. Kolmas peatükk keskendub valminud rakenduse kirjeldamisele.

Siinses töös on kasutatud ChatGPT o3 tekstiroboti [4] abi teksti silumiseks.

Mõisted ja terminid

Mõistete seletused pärinevad AKIT andmekaitse ja infoturbe portaalist [5].

ORM ehk objekt-relatsiooniline kaardistamine - ingl. k. *object-relational mapping*. Programmimehhanism, mis võimaldab andmebaasis andmeid otsida ja käidelda objektipõhistes keeltes.

Raamistik - ingl. k. *framework* -etteantud struktuur, tööriistad ja juhised pakkuv tarkvarakontekst, millele tuginedes saab arendaja kiiremini ning ühtlasemalt luua uusi rakendusi.

Sõltuvus - ingl. k. *dependency*. Viitab tarkvarakomponendile või -teegile, millest mingi teine tarkvaraprojekt või -moodul oma funktsionaalsuse saavutamiseks sõltub.

Relatsioonandmebaas - ingl. k. *relational database*. Andmemudeli tüüp, mis esindab ja salvestab andmeid relatsioonidena ehk seostena tabelite kujul.

Töölauarakendus - ingl. k. *desktop application*. Autonoomne rakendus laua- või sülearvutil vastandina veebipõhisele rakendusele (mida käitatakse brauseriga) või mobiilirakendusele (mis töötab mobiiltelefonil või tahvelarvutis)

Konteinerdus - ingl. k. *containerization*. Tarkvara komplekteerimine koos kõigi ta vajalike lisakomponentidega, näiteks teekidega.

1. Teoreetiline ülevaade

Selles peatükis antakse ülevaade andmebaaside olulisusest rakenduste osana ja miks on vaja andmebaaside õpetamisel tuua näiteid sellest, kuidas andmebaasid suures pildis paigutuvad. Seejärel kirjeldatakse rakenduse arhitektuuri põhimõtteid, keskendudes mitmekihilisele rakenduse arhitektuurile. Viimaks tuuakse välja andmekihi olulisus sellises mitmekihilises rakenduses ja kirjeldatakse, kuidas andmebaasi struktuurid rakenduse käitumist mõjutavad.

1.1 Rakenduse olulisus lisaks andmebaasile

Andmebaasid moodustavad olulise osa kaasaegsest infotehnoloogilisest infrastruktuurist ning enamik inimesi puutub nendega kokku igapäevaste tegevuste kaudu, sageli seda ise teadvustamata. Kuigi otsene kokkupuude andmebaasidega võib jääda märkamata, toimub suur osa kasutajakogemusest andmebaaside vahendusel – näiteks veebipõhiste teenuste, mobiilirakenduste ja sotsiaalmeedia platvormide kasutamisel [6]. Suurem osa tänapäevaseid rakendusi, eriti veebirakendused, tuginevad andmebaasidele kasutajaandmete haldamiseks, dünaamilise sisu loomiseks ning teenuste personaliseerimiseks [7].

Andmete säilitamine on sageli üks olulisemaid eesmärke tarkvararakenduste arendamisel. Paljud rakendused on loodud just selleks, et koguda, hallata ja töödelda andmeid erinevatel otstarvetel, olgu selleks kasutajate haldus, sisu loomine või ärianalüütika [8]. Lisaks andmete säilitamisele on vaja aga andmeid ka kasutajale mugavalt ja arusaadavalt näidata ja võimaldada kasutajal andmete lisamist ja uuendamist.

Arvestades andmebaaside olulistust tänapäevase maailma toimimises, siis on oluline, et andmebaasiõppes ei piirduks vaid teoreetilise käsitlusega, vaid üliõpilastele tuleks näidata ka seda, kuidas andmebaasid toimivad rakenduste osana. See aitab õppijatel mõista, kuidas andmebaasid suhestuvad rakenduse loogika ja kasutajaliidesega ning kuidas tekib andmevoog lõppkasutaja ja andmehoidla vahel [6, 7]. Praktiliste näidete ja rakenduspõhiste ülesannete kaasamine õppetöösse toetab terviklikuma arusaama kujunemist andmebaaside rollist tarkvaraarenduses. See on oluline mitte ainult edasiste õpingute, vaid ka tulevase erialase karjääri seisukohalt [8].

Näiteks võib andmebaasi funktsioonide või protseduuride kasulikkus jääda arusaamatuks, sest nende kasulikkust sõltub rakenduse vajadustest [9]. Sellest tulenevalt võib nende

potentsiaalne kasulikkus õppijatele jääda ebaselgeks, kui pole olemas reaalsed rakendust, mille kontekstis funktsioonide või protseduuride mõju praktikas kogeda. Samuti kujunevad erinevad andmebaasi funktsionaalsused üldjuhul välja vastavalt rakenduse spetsiifilistele nõuetele, mis omakorda määravad nii andmemudeli kui ka süsteemi arhitektuuri kujundamise aluspõhimõtted [6].

Ka õppemotivatsiooni seisukohalt on oluline arvestada Kelleri ARCS-mudelit, mille üks neljast põhikomponendist on *asjakohasus (relevance)* – õppijad on rohkem motiveeritud, kui nad näevad õpitava seoseid reaalse maailmaga [10]. Andmebaaside õpetamisel tähendab see, et õppijad mõistavad paremini õpitava väärtust, kui nad näevad, kuidas andmebaasid integreeruvad rakendustega, mida nad igapäevaselt kasutavad.

Rakenduse ja andmebaasi elutsüklid on tihedalt omavahel seotud ning mõjutavad teineteist kogu süsteemi arendus- ja hooldusperioodi vältel. Andmebaasi disain on sageli üks esimesi ja määravamaid etappe tarkvararakenduse arendusprotsessis, kuna see kujundab viisi, kuidas andmeid struktureeritakse, talletatakse ja kuidas rakendus nendega suhtleb [6, 8].

Kvaliteetne andmebaasistruktuur on oluline süsteemi tõhususe, skaleeritavuse ning andmete terviklikkuse tagamiseks [7]. Halvasti läbimõeldud andmebaas raskendab süsteemi arendamist ja hooldamist, põhjustades tihti vigu, jõudlusprobleeme ning piirates funktsionaalsuse laiendamist. Seetõttu on soovitatav käsitleda andmebaasi ja rakenduse disaini paralleelsete protsessidena, tagamaks nende omavahelist loogilist kooskõla ja süsteemi üldist toimivust [7].

Kokkuvõtvalt võib öelda, et andmebaasid on kaasaegse infotehnoloogilise taristu kese, mida kasutatakse igapäevaselt pea kõigis veebirakendustes ja mobiililahendustes. Nende roll rakenduste jõudluse ja kasutajamugavuse tagamisel on ülioluline, kuid jääb tihti kasutajale märkamata, mistõttu vajavad õppurid andmebaaside õppimisel enam kui vaid teoreetilist lähenemist. Nähtav reaalne kontekst aitab mõista andmemudelite, indeksite, SQL-funktsioonide ja rakendusspetsiifiliste funktsionaalsuste tegelikku tähendust, mis tõstab omakorda ka õpimotivatsiooni.

1.2 Rakenduse arhitektuuri mõiste

Eelnevas lõigus näitasime, miks praktilised kasutusnäited on andmebaaside õpetamise juures olulised. Järgnevalt räägime olulistest aspektidest sellise rakenduse loomise juures.

Rakenduse loomisel tuleb esmalt keskenduda süsteemi arhitektuurile. Kuigi tarkvara arhitektuuri mõiste on mitmeti tõlgendatav ega allu ühesele definitsioonile, võib seda üldjoontes käsitleda kui tarkvarasüsteemi komponentide kõrgeima abstraktsiooni taseme kirjeldust ehk nende disainiotsuste kogumit, mille muutmine on hilisemas arengujärgus kulukas ja keerukas [11].

Tarkvara arhitektuuri mõiste on mitmetähenduslik ka seetõttu, et tarkvarasüsteemi komponente on võimalik mitut moodi defineerida. Komponente võib vaadata kui üksteisest eraldi seisvaid teenuseid, mis võivad asuda näiteks füüsiliselt erinevates serverites. Fowler eristab näiteks mõisteid kiht (*layer*) ja tasand (*tier*), kus tasandite vahel on füüsilised piirid ja kihtide vahel mitte [11].

Selles töös mõistetakse rakenduse arhitektuuri all ühe tervikliku rakenduse ülesehitust. Tarkvarasüsteemi arhitektuuri all on selles töös mõeldud aga omavahel füüsiliselt eraldatud komponentide omavahelisi suhteid, näiteks brauseri ja serveri omavahelist eraldatust.

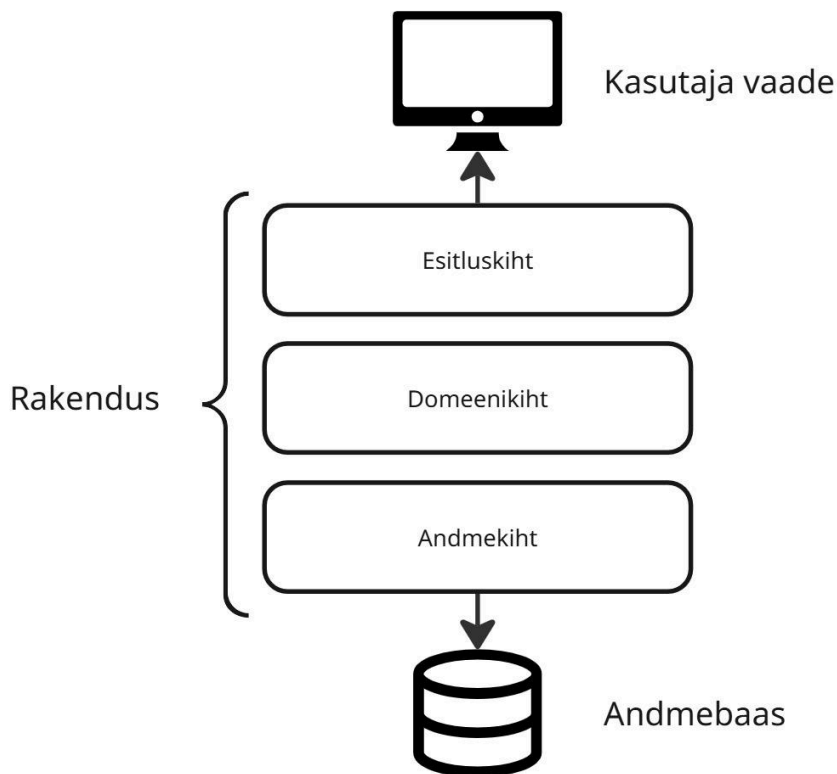
1.3 Mitmekihilise rakenduse arhitektuuri põhimõtted

Kuna kihiline arhitektuur on laialt levinud ja inimesele intuiitiivselt hästi mõistetav [12], siis järgnevalt käsitletakse lähemalt kihilise rakenduse põhimõtteid. Kihiline arhitektuur tähendab rakenduse jagamist kihtideks, kus iga kiht toetub endast allpool oleva kihi peale ehk kasutab allpool oleva kihi funktsionaalsusi, kuid ei sõltu endast ülevalpool olevatest kihtidest. Tavaliselt ei näita ükski kiht ülemistele ka endast allpool olevaid kihte [11].

Sellisel kihtideks jaotamisel on mitmeid eeliseid. Igat kihti saab vaadata kui eraldiseisvat tervikut, ilma et oleks vaja süvitsi mõista teiste kihtide toimimist. Selline lähenemine lihtsustab süsteemist arusaamist, funktsionaalsuste arendamist ja hooldamist [11]. Igat kihti saab lihtsalt asendada ka samalaadse funktsionaalsusega, mis on lihtsalt teistmoodi implementeeritud [ibid.]. Kihtide eraldi käsitlemine loob aluse ka nende peale kõrgema taseme teenuste loomisele. Sellisel juhul ei pea iga uue teenuse arendamise puhul alustama uuesti kõige madalamast kihist [ibid.]. Rakenduse kihtideks jagamine on ka hea koht kust alustada, kui kõik arhitektuurilised otsused ei ole veel tehtud ja rakenduse struktuur võib mõneti muutuda [12].

Süsteemi kihtide arv ei ole kuidagi piiratud ning kohustuslikke kihte ei ole olemas [12]. Enamikul juhtudel koosnevad sellised kihiliselt ehitatud süsteemid aga kolmest või neljast

kihist, väiksemate rakenduste puhul on kaks alumist kihti tavaliselt üheks sulandunud [ibid]. Siinses töös keskendutakse kolmekihilisele arhitektuurile. Kolmekihiline süsteem hõlmab endas esitlus-, domeeni- ja andmekihti [11]. Joonisel 1 on toodud näide kolmekihilisest tarkvarasüsteemi arhitektuurist.



Joonis 1. Näide kolmekihilisest arhitektuurist

Järgnev lõik põhineb Fowleri raamatul [11]. Esitluskiht tegeleb andmete kasutajale loetavale kujule teisendamise ja vastupidi kasutaja sisestatud andmete alumistele kihtidele edastamisega. Rakenduse esitluskiht on tänapäeval enamasti graafiline, ning töötab veebibrauseris, kuid esitluskiht võib olla ka lihtne käsurea rakendus. Veebirakenduse puhul asub esitluskiht tavaliselt kasutaja masinas ja jookseb kasutaja veebibrauseris. Ülejäänud kihid asuvad aga serverarvutis, millega kasutaja masin enamasti HTTP protokolliga suhtleb.

Domeenikiht ehk domeeniloogika või ärioloogika tegeleb andmete ümberarvutamise, valideerimise ning vastavalt esitluskihist saadud juhistele ning eeldefineeritud ärireeglitele andmekihist andmete pärimisega. Seega selles kihis paikneb suurem osa rakenduse mõttest ja funktsionaalsusest [11].

Andmekiht suhtleb süsteemidega, mis rakenduse andmeid endas hoiavad või andmeid muud moodi süsteemi sisse toovad. Andmeid tuleb süsteemi sisse ka läbi esitluskihi, kuid antud juhul ei ole siiski tegu andmekihi alamklassiga. Martin Fowler selgitab erinevust järgmiselt: “Esitluskiht on väline liides pakkumaks teenuseid teistele [...]. Andmekiht on väline liides asjadele, mis sulle teenuseid pakuvad” [11:21]. Tavaliselt on selleks väliseks süsteemiks andmebaas, kuid andmeid saab pärida ka näiteks teistelt rakendustelt [11].

Kihilise rakenduse arhitektuuri suurim eelis on lihtsus ja intuiitiivsus, mistõttu sobib see väikeste projektide jaoks väga hästi. See on ka hea muster, kust alustada, kui kõik arhitektuurilised valikud ei ole veel kindlaks tehtud, seega ajakriitiliste projektide puhul hoiab see kindlasti aega kokku [12].

1.4 Andmebaas süsteemi osana

Andmekihi ülesanne on suhelda väliste teenustega andmete saamiseks või välja saatmiseks. Enamasti on selleks teenuseks andmebaas. Andmebaasi mudelid on palju erinevaid kuid relatsioonandmebaas on kõige populaarsem mudel andmete hoiustamiseks ning SQL on enimlevinud keel relatsioonandmebaasidega suhtlemiseks [13]. Andmebaasidega suhtlemiseks saab kasutada SQL-päringuid, mida arendaja ise kirjutab ning mida rakenduse kood andmebaasi peal käivitab. SQL-päringute loomist õpetatakse ka Tartu Ülikooli kursusel “Andmebaasid LTAT.03.004” [1] ning selles töös eelistatakse sellist lähenemist. Objektipõhiste keelte jaoks on loodud ka objekt-relatsioonilise kaardistamise vahendeid, mis genereerivad SQL-päringuid lähtudes programmikoodist [11].

Objekt-relatsioonilise kaardistamise tööriista ehk ORM-i kasutamise suurimaks eeliseks on arendusmugavus ning arendusprotsessi kiirendamine. ORM-i kasutamine võimaldab arendajatel keskenduda kõrgema taseme loogikale, ilma et nad peaksid liialt muretsema andmekihi implementatsiooni pärast [14]. ORM-id on igati sobilikud väiksemate ja vähemkeerulise andmemudeliga rakendustele, kuid võivad tekitada jõudlusprobleeme, kui rakendus suuremaks kasvab [15, 16]. ORM-id suudavad genereerida lihtsamaid päringuid aga vähegi keerulisemate päringute jaoks on siiski vaja inimese sekkumist [15]. Selles töös valminud rakenduses andmebaasiga suhtlemiseks ORM-e ei kasutata, kuna nende tööriistade kasutamine lisaks lisakeerukust ning üliõpilased omandavad alles objektorienteeritud programmeerimise põhimõisteid.

SQL on seega oluline tööriist integreerimaks rakendust andmebaasiga. Järgnev materjal põhineb Elmasri ja Navathe raamatul [6]. SQL ehk *structured query language* on välja kujunenud standardseks keeleks relatsioonandmebaasidega suhtlemiseks. Just standardiseerimine, mida on eest vedanud ISO (*International Organization for Standardization*) ja ANSI (*American National Standards Institute*), on olnud oluline faktor selle keele ja relatsioonandmebaaside kujunemisel populaarseimaks andmete salvestamise mudeliks. SQL süntaks võib olla küll veidi erinev olenevalt, millist andmebaasi haldustarkvara kasutatakse, kuid põhifunktsionaalsused on kõigil samad.

SQL võimaldab luua funktsioone ja protseduure, et programmikoodi lihtsustada ja sealt andmebaasi spetsiifilisi asju välja tuua. Nende väljatöötamisel lähtutakse reeglina konkreetse rakenduse vajadustest, kuna just rakenduse spetsiifilised nõuded määravad, milliseid operatsioone andmebaasis sooritada ning kuidas ärioloogikat kõige tõhusamalt hallata [6].

Üliõpilastele on oluline seega näha, kuidas rakendus funktsioone kasutab. Üliõpilased näevad andmebaase enamasti veebirakendustes, seega ühe veebirakenduse loomine annab õppuritele hea konteksti õpitava paremaks mõistmiseks [2].

1.5 Andmebaaside õpetamine Tartu Ülikoolis

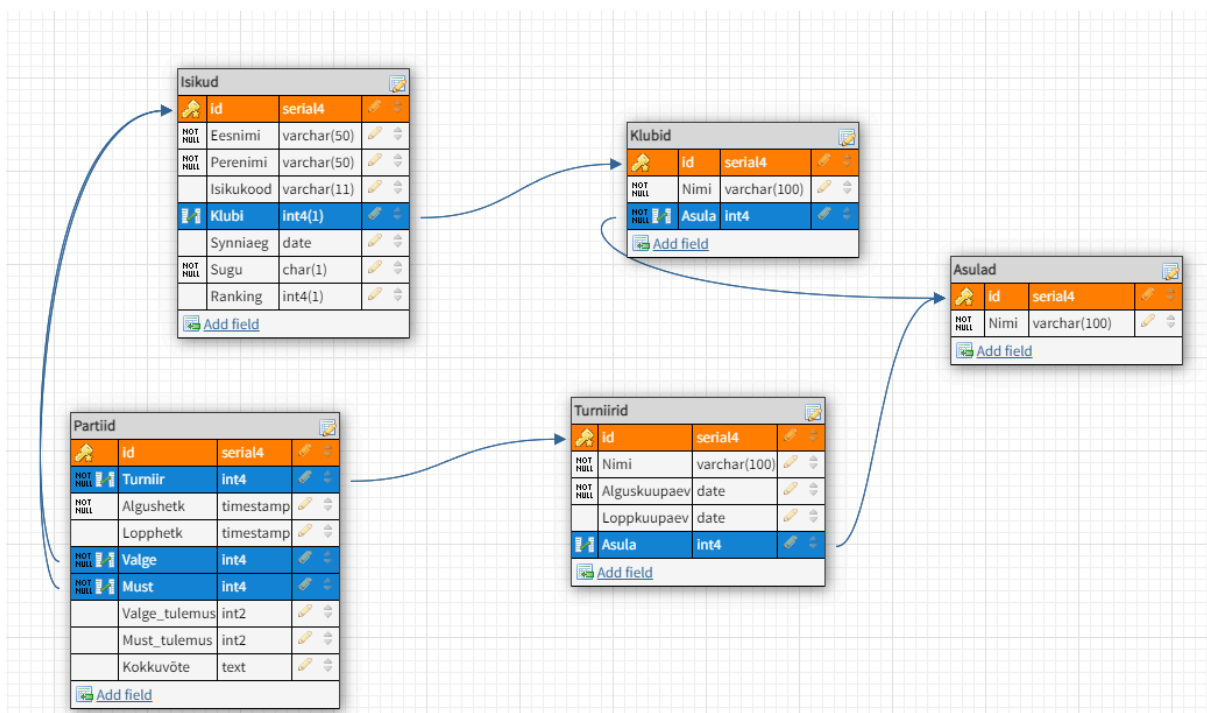
Tartu Ülikooli kursus “Andmebaasid (LTAT.03.004)” [1] kuulub Informaatika bakalaureuseõppekava kohustuslike ainete hulka, ning see läbitakse tavaliselt esimese õppeaasta kevadsemestril. Kursusel osaleb igal aastal tavaliselt umbes 350 õppijat, kusjuures kõik ei õpi Informaatika õppekaval. Kursuse “Andmebaasid” vastutav õppejõud, professor Piret Luik, on ka käesoleva bakalaureusetöö juhendaja.

Kursus on väärt 6 EAP-d ning kestab 16 nädalat, mille jooksul tuleb osaleda loengutes, teha enesekontrolliteste loengumaterjalide kohta, osaleda praktikumides, sooritada 6 koutööd, 3 kontrolltööd, ja rühmatöö. Kursus lõpeb arvestuspäringute ja kirjaliku eksamiga. Arvestuspäringud kontrollivad, kas õppurid on omandanud oskused SQL-ga töötamises ning eksam kontrollib üliõpilaste teoreetilisi teadmisi.

Kesksel kohal kursuse korralduses on nn. male andmebaas, mille peal on üliõpilastel võimalik harjutada SQL-päringute kirjutamist, andmebaasi funktsioonide, protseduuride ja indeksite loomist jne. Male andmebaasi ülesanne on olla mänguväli, kus üliõpilased saavad harjutada andmebaasidega töötamist ning igal õppuril on sellest andmebaasist oma koopia,

millega nad töötavad. See andmebaasi koopia võib olla erinevatel üliõpilastel erinev, s]ltuvalt sellest kui tihti või korrektselt nad oma kodutöid teevad, ning seetõttu on andmebaasis hoitud funktsioonid ja protseduurid näiteks mõnel puudu või valesti tehtud.

Male andmebaas koosneb neljast tabelist, kusjuures kursuse käigus lisandub juurde ka viies tabel. Tabelid kirjeldavad maleklubide, -mängijate, -turniiride ja -partiide omavahelisi suhteid. Sellist andmebaasi võiks näitlikult kasutada maleturniiridel arvestuse pidamiseks. Mängijate puhul salvestatakse näiteks ELO reitingut [17], klubide puhul seda, kes klubisse kuuluvad, turniiride puhul partiisid, mis turniiril mängiti jne. Andmebaasi mudel on kujutatud joonisel 2.



Joonis 2. Male andmebaasi mudel. Allikas: TÜ kursuse “Andmebaasid” Moodle [18]

Karel Paan on teinud aine “Andmebaasid” jaoks samuti ühe näitlikustava rakenduse [19]. See rakendus on aga teistsuguse suunitlusega ja keskendub rohkem sellele, kuidas kasutada ORM-e andmebaasiga suhtlemiseks. Samuti ei kasuta Paani lahendus male andmebaasi. Konkreetselt aines kasutatavat male andmebaasi näitlikustav rakendus on puudu.

2. Rakenduse loomine

Selles peatükis kirjeldatakse nõudeid loodavale rakendusele, seejärel räägitakse tehnoloogilistest valikutest rakenduse arendamisel, kirjeldatakse valminud rakenduse arhitektuuri ning lõpuks arendamise protsessi.

2.1 Nõuded rakendusele

Rakenduse nõuded kirjeldavad mida rakendus tegema peaks. Nõuded jagunevad funktsionaalseteks ja mittefunktsionaalseteks nõueteks [20]. Rakendus on loodud aine “Andmebaasid” vastutava õppejõu kirjeldatud funktsionaalsete nõuete järgi. Koostöös aine õppejõuga seati paika ka mittefunktsionaalsed nõuded.

2.1.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded kirjeldavad, milliseid teenuseid rakendus pakkuma peab, kuidas ta kindlatele sisenditele peab reageerima ja kuidas süsteem kindlates olukordades käituma peab [20].

Funktsionaalsete nõuete täitmine või mittetäitmine sõltub iga üliõpilase puhul ka sellest, kas tema andmebaasis on olemas kõik vajalikud funktsioonid ja vaated, mida rakendus kasutab.

Kuna tegu on õppevahendiga, siis on rakenduse kasutaja defineerimine kahetimõistetav. Rakenduse kasutaja võib olla kas üliõpilane, praktikumijuhendaja või keegi kujuteldav maleturniiri korraldaja. Kasutaja all mõistame funktsionaalsete nõuete kirjeldamisel üliõpilast, kes seadistab rakenduse enda arvutis jooksma.

Selle töö käigus valminud rakenduse funktsionaalsed nõuded on järgnevad:

- 1) Kasutaja peab saama vaadata, kes on kui palju võitnud ja kaotanud.
- 2) Kasutaja peab saama näha mängijate kohta andmeid, sealhulgas mängija sugu, vanust, klubi, võite ja kaotusi.
- 3) Kasutaja peab saama näha andmeid klubide kohta, sealhulgas kus antud klubi asub, ja mis ränkingud klubi mängijatel on.
- 4) Kasutaja peab saama näha, kes võidab enam mängides õhtul, kes hommikul.

- 5) Kasutaja peab saama näha andmeid partiide kohta, sealhulgas kes mängisid ja kes partii võitis.
- 6) Kasutaja peab saama lisada turniire ja nendes partiisid.
- 7) Kasutaja peab saama näha andmeid turniiride kohta, sealhulgas turniiri toimumise kohta ja turniiril mängitavaid partiisid.

2.1.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded seavad piirid süsteemi poolt täidetavatele funktsionaalsustele ja pakutavatele teenustele. Need on näiteks ajalised piirangud, piirid arendusprotsessile ja standarditest tulevad piirangud [20]. Antud kontekstis kirjeldavad rakenduse mittefunktsionaalsed nõuded ka rakenduse toimimist õppevahendina.

Rakendust kasutatakse kahte moodi: esiteks iga üliõpilase enda masinas, kus igaüks saab selle ise käima panna ja seal muudatusi teha. See on ennekõike neile, kes kavatsevad boonusülesannet lahendada. Kuna iga kursust läbiv üliõpilane eeldatavasti boonusülesannet täita ei soovi, siis on loodud ka pilveteenuses jooksumise variant, kus õppurid saavad ilma seadistamata rakendust lihtsalt uurida.

Selle töö käigus valminud rakenduse mittefunktsionaalsed nõuded on järgnevad:

- 1) Rakendus peab olema lihtsasti ehitatav ja käivitav iga üliõpilase enda masinas.
- 2) Rakendust peab olema võimalik üliõpilastel muuta.
- 3) Rakendus peab jooksuma pilveteenuses, et seda oleks mugav demonstreerida, ilma, et peaks ise rakendust üles seadma.

2.2 Rakenduse ülesehitus

Rakenduse arendamiseks on mitmeid võimalusi – seda saab teostada kas töölauarakendusena, mobiilirakendusena või veebirakendusena, mis töötab kasutaja veebilehitsejas. Käesoleva bakalaureusetöö raames loodud rakendus on veebipõhine. Veebirakenduse üheks oluliseks eeliseks on asjaolu, et kasutaja ei pea eraldi tarkvara paigaldama, eeldusel et tal on juba olemas veebilehitseja. On ka toodud, et üliõpilased näevad tavaliselt andmebaase veebirakenduste taustal töötamas, mistõttu on just veebirakendused relevantsemad [2].

Veebirakendus võib olla täielikult staatiline või sisaldada erineval määral dünaamilisi elemente. Staatilise veebilehe näiteks võib tuua isikliku kodulehe, mis edastab igal külastusel sama sisu, sõltumata kasutaja sisendist – tema serveritav sisu on staatiline, st. sisu ei muutu. Dünaamilised veebilehed seevastu genereerivad sisu vastavalt kasutaja sisestatud andmetele – näiteks otsinguvälja kasutamisel kuvatakse tulemused vastavalt sisestatud päringule [21]. Käesolevas töös valminud rakendus kuulub dünaamiliste veebirakenduste hulka.

Tehnoloogiate valimisel on tugevalt lähtunud põhimõttest, et üliõpilased peavad suutma rakendust laiendada, et boonusülesannet täita. Seetõttu on valitud tehnoloogiaid populaarsuse alusel. Rakenduse loomiseks kasutatud tehnoloogiaid tutvustatakse ka Tartu Ülikooli kursusel “Veebirakenduse loomine (LTAT.05.004)” [22], mis jätab võimaluse, et rakendus tutvustab neile tehnoloogiaid, mida hiljem sügavuti õpetatakse. Samuti on need tehnoloogiad töö autori jaoks kõige mugavamad kasutada.

Andmekihi funktsiooni täidab Node.js server, mis suhtleb otseselt andmebaasiga. Node.js on avatud lähtekoodiga JavaScripti käituskeskkond, mis võimaldab JavaScripti kasutamist serveripoolsete rakenduste arendamisel. See põhineb Google Chrome'i V8 JavaScripti mootoril ning võimaldab luua tõhusaid ja skaleeritavaid võrgurakendusi [21].

Kuna domeenikiht on tihedalt seotud rakenduse ärioloogikaga [15] ja selles töös loodavas rakenduses on ärioloogikat väga vähe, ei ole domeenikiht rakenduses selgelt eristatud. Selle asemel on domeenikiht suurel määral integreerunud esitluskihiga. Näiteks domeenikihile omased funktsioonid, nagu ärireeglite rakendamine uue mängija lisamisel, on realiseeritud otse esitluskihis paikneva loogika kaudu.

Kasutaja brauseris jookseb esitluskiht, mis esitleb serverilt saadud andmeid. Selle jaoks on kasutatud Vue.js raamistikku. Vue.js on JavaScripti raamistik veebirakenduste kasutajaliideste loomiseks [23]. Töö autor otsustas kasutada Vue.js raamistikku, kuna see on populaarne ja hästi dokumenteeritud, mistõttu on probleemidele lahenduste leidmine lihtne. Sarnastest raamistikest on StackOverflow küsimustiku põhjal React ja Angular veel populaarsemad [24], kuid autori jaoks on Vue.js kasutamine mugavam. Vue.js raamistiku kasutamist õpetatakse ka Tartu Ülikooli kursusel “Veebirakenduse loomine”, seega see on tõenäolisemalt midagi, millega üliõpilased on varem kokku puutunud, täites seega tõenäolisemalt nõuet, et rakendust peab olema lihtne üles seada ja laiendada.

Vue.js-põhine lähtekood ei ole brauseris vahetult täidetav; enne kasutajale edastamist tuleb see transpileerida ning koondada standardseteks HTML-, CSS- ja JavaScript-ressurssideks. Ehitustööriist Vite genereerib Vue-komponentidest optimeeritud staatilised lehed, mida saab teenindada tavapärase veebiserveri (nt Apache HTTP Server) kaudu [25]. Vite'i kasutegurid ilmnevad eeskätt selle äärmiselt kiire kompileerimistsükli ja minimaalse konfiguratsioonivajaduse tõttu, mis asetavad tööriista soodsasse võrdluse teiste lahendustega, nagu näiteks Webpack, ning parandavad märkimisväärselt arendajate töövoos efektiivsust [25, 26].

Rakenduse loomisel Vue.js raamistikuga kasutati lisaks ka Vuetify teeki. Vuetify on komponentide kogum, mis sisaldab hulgaliselt eeldefineeritud elemente, mida on võimalik rakenduses mugavalt kasutada ja kohendada [27]. Selliste komponentide hulka kuuluvad näiteks tabelid andmete esitamiseks, konteinereid teiste komponentide jaotamiseks ning otsingukastid, mis võimaldavad rakenduse funktsionaalsust ja kasutajaliidest paindlikult laiendada [ibid.].

Arendusprotsessi hõlbustamiseks integreeriti mitmeid teekes, sealhulgas Axios, Express.js, node-psql ja unplugin-vue-router. Teekide valimisel on lähtutud põhimõttest, et teek oleks piisavalt populaarne ning hästi dokumenteeritud. Teeke valitakse mingi kindla probleemi lahendamiseks, nagu näiteks andmebaasiga suhtlemine või veebirakenduse navigatsiooni korraldamine.

Unplugin-vue-router võimaldab veebirakenduse navigatsiooni korraldada failistruktuuri põhiselt, mistõttu piisab näiteks mängijate Vue.js komponendi lisamisest mängijate-nimelisse kausta, et suunata kasutaja automaatselt vastavale lehele [28]. Axios, mis on lubadustepõhine (promise-based) HTTP klienditeek, võimaldab Vue.js-rakenduse sees hõlpsasti teostada serveri vastu päringuid, tagades ühtlasi ülevaatliku koodistruktuuri ning parema tõrke- ja tulemustöötluse [29].

Serveri pool on kasutatud express.js raamistikku, mis teeb lihtsaks serveri otspunktide haldamise. Express.js on laialt kasutatav Node.js-raamistik, mis võimaldab luua paindlikke ja mastaapseid veebirakendusi, toetudes rikkalikule ökosüsteemile [30]. Express.js on raamistik REST tüüpi liideste loomiseks. Andmebaasiga suhtlemiseks on kasutatud ka node-postgres teeki, mis suhtleb andmebaasi serveriga. Node-postgres lihtsustab PostgreSQL andmebaasidega suhtlemist, pakkudes struktureeritud viisi päringute koostamiseks ja

haldamiseks [31]. See võimaldab kirjutada lihtsalt SQL päringuid ja teek korraldab päringute andmebaasile edastamise ja tulemuste näitamise [ibid.].

Et täita rakendusele seatud mittefunktsionaalseid nõudeid, eriti just nõuet, et rakendus peab olema lihtsasti üles seatav, jookseb rakendus ka konteinersüsteemis Docker. Docker on StackOverflow küsitluse põhjal kõige populaarsem omasarnane tööriist [24], seega on see kõige tõenäolisemalt üliõpilastele juba varasemalt tuttav. Konteinersüsteemil Docker on ka üsnagi mugav kasutajaliides, mis teeb ta algajale sõbralikumaks. Seetõttu ei tahetud kasutada alternatiive, nagu näiteks podman või containerd. Iga üliõpilane saab seada rakenduse üles paari käsuga, juhul kui Docker on nende arvutis juba olemas.

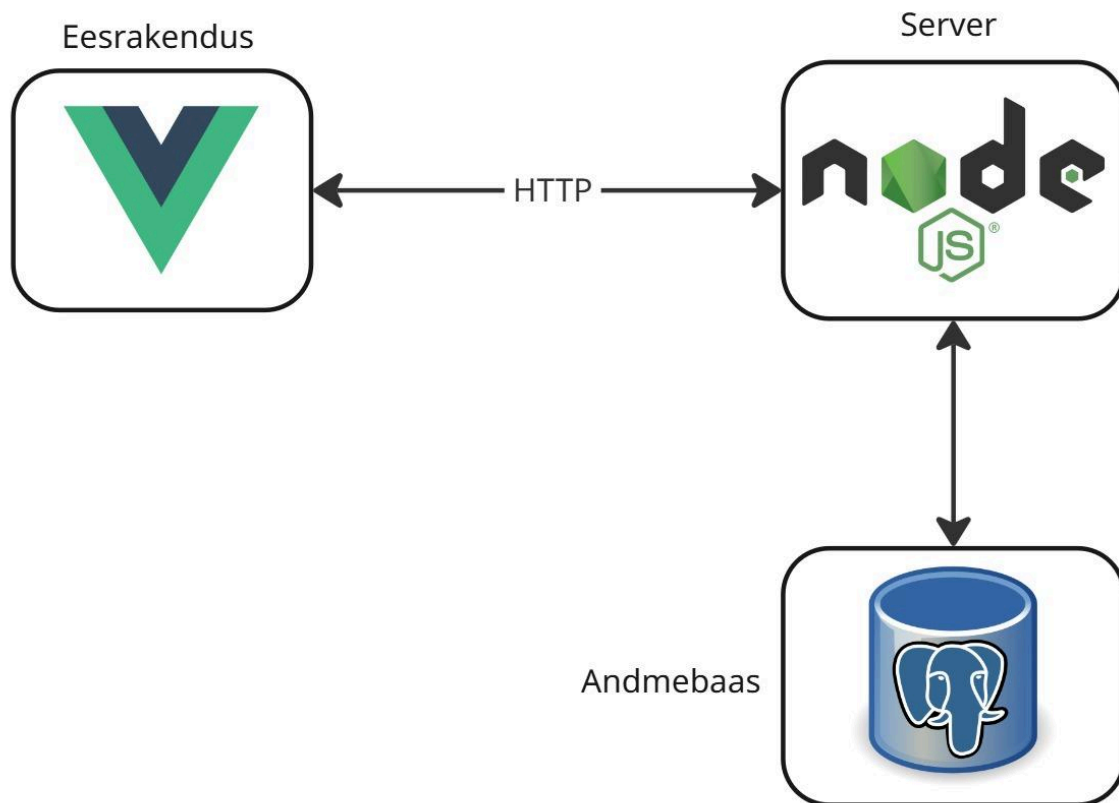
Docker on konteineriseerimise tarkvara, mis võimaldab rakendusi pakkida väikesesse virtuaalmasinasse, mida nimetatakse konteineriks. Konteineris jookseb oma operatsioonisüsteem ja keskkonnamuutujad ja ta on kasutaja arvutist isoleeritud. See võimaldab rakendust igas masinas samas keskkonnas jooksutada, mis teeb rakenduse testimise lihtsamaks, kuna lubab eeldada, et rakendus jookseb alati samas keskkonnas ja vähendab jällegi konfigureerimise vajalikkust. Selle bakalaureusetöö käigus valminud rakenduse kasutajaliidest serveriv protsess, server (*backend*) ja andmebaasiserver on kõik eraldi konteinerid, kuna konteinerisüsteemi Docker dokumentatsioon soovib käivitada ühes konteineris ühte protsessi [32].

Docker Compose on konteinerite orkestreerimise tööriist, mis võimaldab mitmest konteinerist koosneva rakenduse koordineeritud käivitamist ja haldamist. Ühtse Compose konfiguratsioonifaili kaudu saab iga konteineri jaoks deklareerida keskkonnamuutujad, määratleda omavahelised võrgusidemed ning seadistada muud taristuparameetrid, luues seeläbi loogiliselt integreeritud teenuste kompleksi. Selline deklaratiivne lähenemine võimaldab kogu konteineripõhise süsteemi käivitada või peatada ühe käsuga, mis lihtsustab juurutamist ja vähendab operatiivset keerukust. Compose moodul käib standardelt Dockeriga kaasa, mistõttu ei nõua see eraldi tarkvarakomponentide paigaldamist [33].

2.3 Valminud rakenduse arhitektuur

Eesrakendus (*front-end*) vahetab serveriga andmeid HTTP-protokolli vahendusel, samal ajal kui server edastab päringuid andmebaasile. Seejuures laaditakse vajalikud andmed brauserisse dünaamiliselt vastavalt kasutaja tegevusele. Näiteks tuuakse mängijatega seotud

andmed sisse alles siis, kui kasutaja avab vastava lehe. Andmeid värskendatakse iga kord kui lehte värskendatakse, et tagada ajakohase info kuvamine. Rakenduse komponentide omavahelised suhted on toodud joonisel 3.



Joonis 3. Rakenduse komponentide arhitektuur

Kuna kõik üliõpilased ei soovi boonusülesannet lahendada, kuid rakenduse demonstreerimine on siiski oluline, on rakendus üles seatud ka pilves, kasutades tasuta pilveteenuseid: Vercel [34] kasutajaliidese (*frontend*), Render [35] taustateenuste (*backend*) ning Supabase [36] andmebaasi majutamiseks. Need teenused on valitud just seetõttu, et nad on tasuta, alternatiivid vajavad kas tasulist plaani või pakuvad tasuta majutamise võimalust vaid mingiks kindlaks perioodiks. Selline lahendus võimaldab õppejõududel praktikumide käigus rakendust mugavalt demonstreerida. Siiski tuleb arvestada teatud piirangutega – nii Render kui ka Supabase puhul lõpetatakse rakenduse aktiivsena hoidmine, kui seda pikemat aega ei kasutata [35, 36]. See omakorda põhjustab esmakordsel taasavamisel viivituse, kuna teenus tuleb esmalt uuesti käivitada. Seetõttu ei ole tegemist ideaalse lahendusega, kuid see sobib siiski hästi demonstreerimise eesmärgi saavutamiseks.

Veebilehe esmasel laadimisel edastab veebiserver brauserile kolme tüüpi ressursid: HTML-, CSS- ja JavaScript-failid. HTML-dokument määratleb kasutajaliidese semantilised elemendid ja nende loogilise struktuuri; CSS-stiliseerimisreeglid kirjeldavad nende elementide esitlust, sealhulgas paigutust, tüpograafiat ja visuaalseid efekte; JavaScript-i käituskood lisab lehele käitumusliku mõõtme, võimaldades kasutajapoolset interaktsiooni nagu rippmenüüde avamine, andmete reaajas filtreerimine või muu dünaamiline sisu värskendamine [37].

Antud veebirakendus kasutab täielikult dünaamilist andmeedastust: esmasel päringul saadab veebiserver brauserile vaid HTML-, CSS- ja JavaScript-ressursid, mille järel kliendipoolne JavaScript algatab täiendavad päringud taustateenustesse andmete toomiseks. Staatiline esitus sisaldab üksnes pealkirju ja teisi struktuurseid konteiner-elemente, kuid kogu sisuline teave kuvatakse alles pärast vastavate andmete saabumist. Kui andmeid ei laeku, jäävad vastavad kasutajaliidese komponendid üldse kuvamata.

Selles töös valminud rakenduse puhul on tegu SPA tüüpi rakendusega. SPA ehk üheleherakendus (*single page application*) kujutab endast veebiarhitektuurilist paradigmat, mille korral kogu kasutajaliidese struktuur laaditakse brauserisse üheainsa algpäringu käigus; järgnevalt piirdub kliendi- ja serverivaheline suhtlus vaid valikuliste andmepäringutega [38]. Seetõttu ei hangita kasutuse käigus täiendavaid HTML-, CSS- ega JavaScript-ressursse, vaid üksnes domeenispetsiifilise sisuga andmeid, mis renderdatakse dünaamiliselt olemasoleva lehekülje kontekstis [ibid.]. Antud malerakenduse kontekstis laetakse näiteks alguses klubide leht, siis küsitakse serverist andmeid klubide kohta, ning renderdatakse klubide loetelu vastavalt serverilt saadud vastusele.

Uuringud näitavad, et üheleheküljelise rakenduse peamine väärtus on kasutajakogemuse optimeerimine: kogu kasutajaliidese struktuur laaditakse brauserisse esmasel päringul ning edasised interaktsioonid kutsuvad esile vaid asünkroonseid andmepäringuid, mis välistavad täislehe uuesti laadimise [38]. See arhitektuur kiirendab vaatevahetust, kõrvaldab traditsioonilise lehe “vilkumise” ning tagab katkematu, töölauarakendusele sarnaneva navigeerimiskogemuse [39]. Sellise üheleherakendusena on vormistatud just rakenduse esitluskiht. Rakenduse lähtekood on toodud ka lisa 1.

2.4 Arendamise protsess

Nagu eelnevalt mainitud, soovitatakse rakenduse loomisel alustada rakenduse arhitektuuri paikaseadmisega [11]. Valminud rakenduse arendamise algusetapis mõeldi esmalt rakenduse kihtideks jagamise peale ning seega valiti tehnoloogiad, mida igas kihis kasutada.

Kui rakenduse arhitektuursed mustrid ja tehnoloogiad olid valitud, ehitati esmalt valmis eesrakendus. Eesrakendus on kogu rakenduse kõige olulisem osa, seal peitub suurem osa kogu rakenduse loogikast. Kuna rakendus on jagatud kihtideks ning kihid on üksteisest loogiliselt eraldatud, saab eesrakendust luua ka ilma tagarakenduse tuge eelnevalt loomata. Eesrakenduse ülesandeks on andmete visualiseerimine, ning andmeid saab arendamise eesmärgil ka otse eesrakenduse sisse kirjutada, need testeemärgil loodud andmekogud hiljem eemaldades.

Seejärel ehitati tagarakendus, mis serveerib eesrakendusele andmeid male andmebaasist ning ühendati tagarakendus eesrakendusega. Nüüd on võimalik näha eesrakenduses korrektseid andmeid, mis tulevad otse male andmebaasist. Tagarakenduse arendamine käis järk-järgult, vastavalt eesrakenduse vajadustele. Näiteks loodi esmalt SQL-päring andmebaasist klubide lugemiseks, seejärel pandi tagarakendus SQL-päringut kasutama ning andmeid ühtsesse vormi panema ja lõpuks loodi eesrakendusse tugi andmete tagarakenduselt küsimiseks ning kasutajale näitamiseks. Sarnaselt käituti kõigi rakenduse funktsionaalsustega.

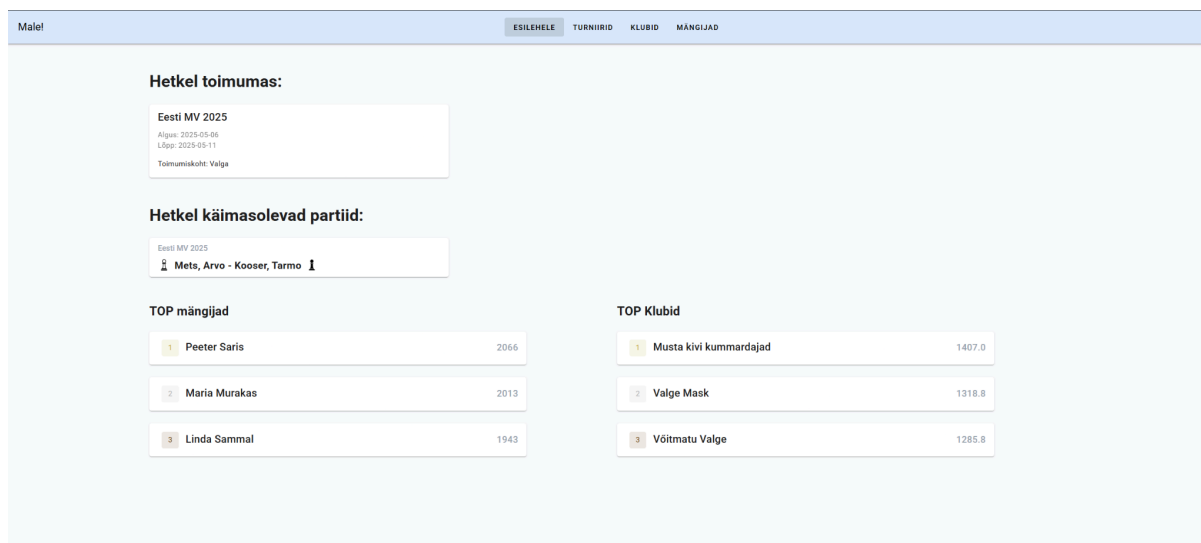
Kokku võttis rakenduse arendamine hinnanguliselt neli töönaalat, millest kolm nädalat kulus rakenduse arendamisele ning üks nädal vigade parandamisele, rakenduse konteinersüsteemis Docker käima sättimisele ning rakenduse pilveteenustesse juurutamisele.

3. Valminud lahendus

Selles peatükis kirjeldatakse lähemalt valminud maleturniiride rakendust, käies üle rakenduse olulisemad funktsionaalsused. Seejärel tutvustatakse ka ülesannet rakendust laiendada, mida üliõpilased saavad boonuspunktide teenimiseks lahendada.

3.1 Maleturniiride rakendus

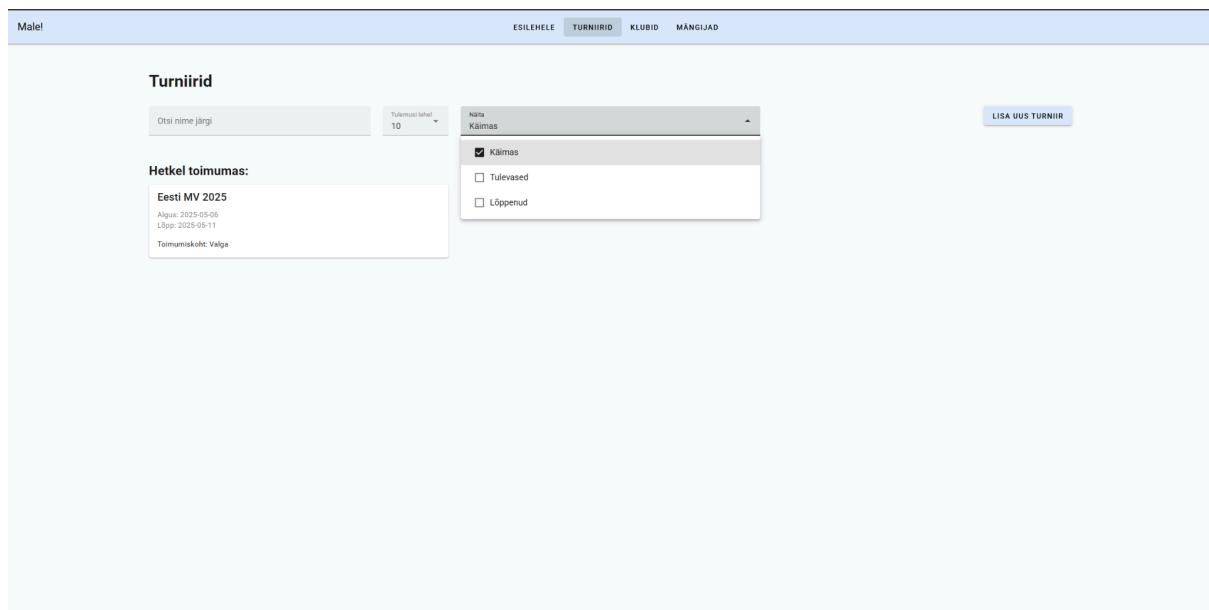
Rakendusel on neli lehte - esileht, turniiride leht, klubide leht ja mängijate leht. Nende lehtede vahel saab kasutaja navigeerida kasutades lehe ülaosas olevat navigatsiooniriba. Lehed on loodud vastavalt rakendusele esitatud funktsionaalsetele nõuetele. Esilehel saab näha üldist informatsiooni maleturniiride, -klubide ja -mängijate kohta. Siin on näiteks loetletud hetkel toimuvad turniirid ja partiid ning mängijate ja klubide edetabel. Esilehte on kujutatud joonisel 4.



Joonis 4. Maleturniiride rakenduse esileht.

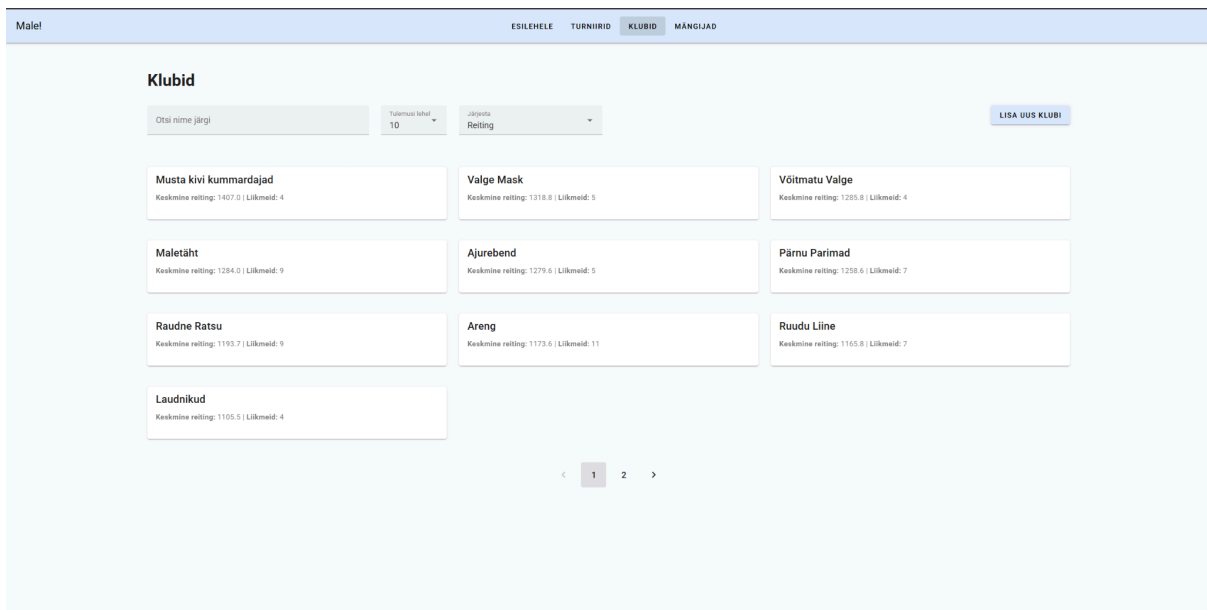
Turniiride lehel kuvatakse rohkem informatsiooni turniiride kohta. Siin saab filtreerida turniire vastavalt sellele, kas turniir on hetkel toimumas, juba lõppenud või alles tulemas. Samuti saab turniire nime järgi otsida. Leht on viidud vastavusse andmebaasis oleva turniiride tabeliga ning peaks tekitama õppijatele äratundmistunnet. Üliõpilased saavad ise oma andmebaasi vastu lihtsa päringu koostada, ning veenduda, et tuleb sama tulemus. See võiks jällegi luua õppijatele konteksti, kuidas andmebaasi rakenduses kasutatud on. Siin lehel saab ka kasutaja turniire juurde lisada. See peaks õppuritele näitlikustama andmete lisamise

mehhanismi, üliõpilased saavad hiljem veenduda, et nende andmebaasi tõepoolest lisati uus kirje, kui vajutada rakenduses “Salvesta” nuppu. Turniiride leht on kujutatud joonisel 5.



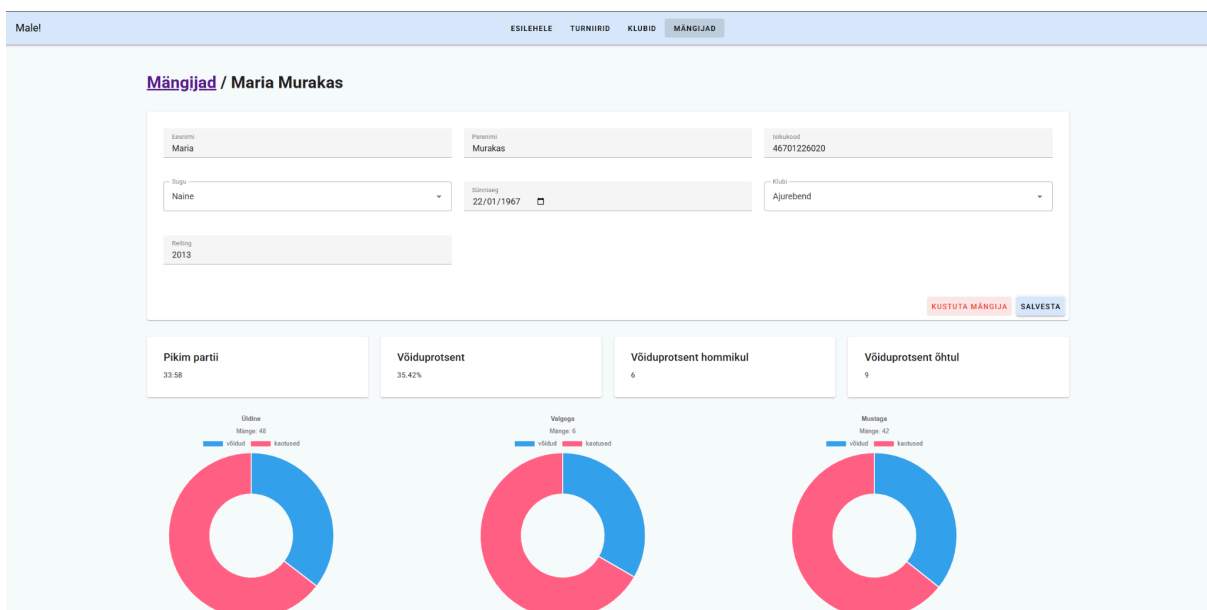
Joonis 5. Maleturniiride rakenduse turniiride leht.

Klubide lehel on informatsioon klubide kohta. Klubide järjestust saab seada kas nime, liikmete arvu või liikmete keskmise ELO reitingu järgi. Klubisid saab ka nime järgi otsida. See leht on samamoodi otseses vastavuses andmebaasis oleva klubide tabeliga ning peaks jällegi aitama õppijatel andmebaase rakenduse osana mõista. Klubide leht on kujutatud joonisel 6.



Joonis 6. Maleturniiride rakenduse klubide leht.

Mängijate leht on vormilt samasugune, nagu on klubide leht. Ka siin saab mängijaid nime ja reitingu alusel järjestada ning nime järgi otsida. Rohkem infot annab kasutajale ühe mängija detailvaate leht, kuhu saab minna klikkides mängijate otsingulehele ühe mängija peale. Siin on graafikuid, näitlikustamaks andmebaasi agregaatfunktsioone nagu näiteks keskmise võtmise. See peaks näitama õppuritele agregaatfunktsioonide olulisust ning tekitama jällegi sügavat arusaama teemast. Siin lehel on võimalik ka mängija andmeid uuendada. Mängija detailvaadet on näidatud joonisel 7.



Joonis 7. Maleturniiride rakenduse mängija detailvaade.

Kokkuvõtvalt võib öelda, et rakendus on koostatud vastavalt eelpool toodud funktsionaalsetele nõuetele, ning rakenduse loomisel on mõeldud sellele, kuidas andmebaasi olulisust rakenduses üliõpilastele välja tuua.

3.2 Boonusülesanne üliõpilastele

Üliõpilastele on koostatud ka ülesanne, mille lahendamise eest on võimalik kursusel teenida boonuspunkte. Ülesanne eeldab rakenduse muutmist, et näidata klubi lehel klubi mängijate edetabelit. Sellise ülesande lahendamine annab üliõpilastele võimaluse praktiliselt oma teadmisi täiendada. Rakenduse muutmine ei ole kursuse “Andmebaasid” kohustuslik osa ning ülesanne on mõeldud boonusena neile, kes soovivad andmebaaside kasutamist rakenduses ise katsetada. Ülesande eesmärk ongi just rakendust muuta, mitte keerulist päringut kirjutada, sest ülesandega tahetakse luua ettekujutus sellest, kuidas andmebaas rakenduse osana toimib.

Ülesande püstitus on järgnev: loo klubi lehele poodium, kus on näha klubi 3 parimat mängijat, kasutades andmete pärimiseks funktsiooni `f_klubiparimad()`. Ülesande kirjeldus on õppuritele kättesaadav rakenduse lähtekoodi repositooriumis ning on kujutatud joonisel 8.

Boonusülesanne

Klubi lehele saab funktsionaalsusi juurde lisada. Üks selline asi oleks näiteks klubi top 3 mängija esile tõstmine.

Lüügu lehele "Klubid" ja vali üks klubi.

See leht näitab juba mängijaid, kes klubis on, kuid oleks vaja juurde tekitada klubi mängijate top 3.

Selleks saab kasutada andmebaasi funktsiooni `f_klubiparimad()`

Ülesanne: loo klubi lehele poodium, kus on näha klubi 3 parimat mängijat, kasutades andmete pärimiseks funktsiooni `f_klubiparimad()`

Lahendus võiks näha välja midagi sellist, nagu on esilehel.

TOP mängijad		
1	Peeter Saris	2066
2	Maria Murakas	2013
3	Linda Sammal	1943

Kui ülesanne on lahendatud, lüüka oma muudatused githubi üles ja esita oma repositooriumi link Moodle keskkonda.

```
git add .
git commit -m "boonusülesanne"
git push
```

Joonis 8. Boonusülesande kirjeldus rakenduse GitHub'i lehel.

3.3 Rakenduse kasutamine õppevahendina

Rakenduse mittefunktsionaalsete nõuete all on nõutud, et rakendust peaks saama kasutada mitut moodi. See on põhjusel, et rakenduse enda arvutisse alla tõmbamist ja püstiseadmist eeldatakse pigem õppuritelt, kes soovivad boonusülesannet lahendada. Rakenduse paigaldamine ei ole seega kõigilt üliõpilastelt nõutav. Selleks, et õppurid saaksid siiski male andmebaasis toimuvaga visuaalselt tutvuda, on loodud ka versioon rakendusest, mis töötab pilveteenuses.

Ühe variandina saab seega üliõpilane enda masinasse tõmmata GitHubis oleva lähtekoodi, seadistada seal ühendusparameetrid oma arvutis oleva male andmebaasiga ning näha konkreetselt enda andmebaasis toimuvat. Kui rakenduse mõni leht ei toimi, siis võib eeldada,

et andmebaasis on midagi katki, näiteks puuduv tabel või funktsioon. Rakendus on loodud kasutama andmebaasi, mis õppuritel kursuse lõpuks olema peaks. Rakenduse ülesseadmine annab seega võimaluse oma andmebaasi korrektsust valideerida. Kui rakendus on paigaldatud, siis saab lahendada ka boonusülesannet, ning teenida ülesande korrektse lahenduse eest kaks boonuspunkti.

Kuna rakenduse üles seadmine ei ole kohustuslik, siis saavad need üliõpilased, kes seda teha ei soovi, näha pilveteenuses olevat rakendust, mida praktikumijuhendajad praktikumis kõigile tutvustavad. Praktikumijuhendajatele oli antud ülesanne rakendust lühidalt tutvustada ning anda praktikumis osalejatele ülesandeks katsetada SQL-päringuid, mis annaksid sarnaseid tulemusi, kui rakenduses kuvatud. Näiteks, kas klubide lehel kuvatav klubide info on sama kui oma andmebaasi vastu tehtud päring.

3.4 Tagasiside rakendusele

Valminud rakenduse hindamiseks koostati tagasisideküsitlused õppijatele ning praktikumijuhendajatele. Küsimustikes paluti hinnata rakenduse ja boonusülesande kasulikkust skaalal 1-5 punkti ning kirjutada mis rakenduse ja boonusülesande juures meeldis ja mida peaks parandama.

3.4.1 Küsimustik praktikumijuhendajatele

Praktikumijuhendajatele edastatud küsimustikus paluti hinnata järgnevaid aspekte 5 punkti skaalal, kus 1 tähendab väitega mittenõustumist ja 5 täielikult nõustumist. Juhendajad said küsimustiku 07.05.2025, peale praktikumis rakenduse tutvustamist. Küsimustik on toodud lisa 2.

- 1) Rakendus aitab luua üliõpilase jaoks paremat ülevaadet, kuidas andmebaase igapäevaelus kasutatakse.
- 2) Usun, et boonusülesande lahendamine aitab üliõpilasel mõista andmebaaside olulisust
- 3) Usun, et oma andmebaasi rakenduse osana nägemine parandab õppijate õppemotivatsiooni.
- 4) Kokkuvõttes on rakendus üliõpilase jaoks kasulik.

Samuti paluti vastata kahele avatud vastusega küsimusele.

- 5) Mis on rakenduses hästi teostatud?

6) Mida tuleks rakenduse juures veel lihvida?

Küsimustikule vastasid 7 praktikumijuhendajat 13st. Kokkuvõtte küsimustiku vastustest on esitatud tabelis 1.

Tabel 1. Kokkuvõtte praktikumijuhendajate tagasisidest

küsimus/hinne	1	2	3	4	5	keskmine
Rakendus aitab luua üliõpilase jaoks paremat ülevaadet, kuidas andmebaase igapäevaelus kasutatakse	0 (0%)	0 (0%)	1 (14.3%)	1 (14.3%)	5 (71.4%)	4.57
Usun, et boonusülesande lahendamine aitab üliõpilasel mõista andmebaaside olulisust	0 (0%)	0 (0%)	1 (14.3%)	2 (28.6%)	4 (57.14%)	4.43
Usun, et oma andmebaasi rakenduse osana nägemine parandab õppijate õppemotivatsiooni	0 (0%)	0 (0%)	0 (0%)	4 (57.14%)	3 (42.9%)	4.43
Kokkuvõttes on rakendus üliõpilase jaoks kasulik	0 (0%)	0 (0%)	1 (14.3%)	2 (28.6%)	4 (57.14%)	4.43

Enamik praktikumijuhendajaid hindasid, et rakendus loob õppuritele paremat konteksti andmebaaside olulisusest tänapäevase maailma toimimises. Keskmine hinne 7 vastuse põhjal on 4.57/5-st, mis on ka kõige kõrgemini hinnatud väide. Positiivse hinde said ka ülejäänud väited, keskmise hindega 4.43/5-st.

Rakenduse juures meeldis juhendajatele just rakenduse visuaalne pool. 5 juhendajat 7-st tõid välja, et rakenduse kasutajaliides näeb hea välja ning on loogiline ja lihtne kasutada. Üks juhendaja tõi ka välja, et just male andmebaasi kasutamine rakenduse osana on positiivne ning loodud vaated on hästi kooskõlas andmebaasi vaadete ja tabelitega.

Rakenduse juures saaks parandada 2 juhendaja hinnangul algse laadimise kiirust. Ühe juhendaja vastuste põhjal on puudu ka lehest, mis tutvustaks lähemalt rakenduse võimalusi. Üks juhendaja arvas, et ülesande püsitus on liiga lihtsakoeline ning ei nõua õppuritelt andmebaasi päringute tegemist ega tutvusta ORM-ide kasutamist. Sama juhendaja arvas samuti, et puudu on kontainersüsteemi Docker kasutamise õpetusest rakenduse ülesseadmise juhendis.

Arvestades valdavalt positiivseid hinnanguid punktiskaalal, hinnangut, et rakenduse kasutajaliides on loogiline ja hea kasutada ning suuremate puuduste puudumist tagasisideküsimustikus antud vastutstes, võime väita, et praktikumijuhendajatele rakendus meeldis ning nad hindavad positiivselt rakenduse kasulikkust kursuse “Andmebaasid” õppekava osana.

3.4.2 Küsimustik üliõpilastele

Kursuse üliõpilastele esitati samuti küsimustik, kus paluti õppuritel vastata kahele küsimusele. Küsimustik on toodud ka lisa 3.

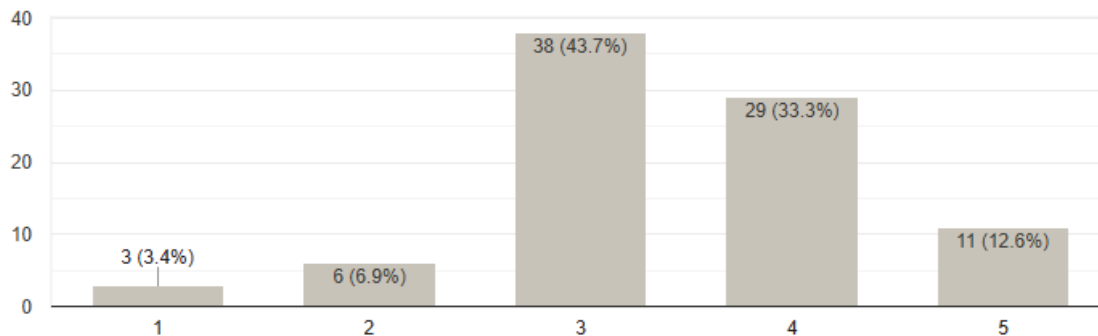
- 1) Kuivõrd aitas viimases praktikumis näidatud Male andmebaasi rakendus mõista, kuidas andmebaase igapäevaelus kasutatakse? (skaalal 1-5).
- 2) Kui lahendasite bonusülesande, milles oli vaja rakendust täiendada, palun kirjutage, mis selle ülesande juurdes teile meeldis või mida annaks paremaks muuta?

Kokkuvõttes hindasid kursusel osalevad üliõpilased rakendust keskmise hindega 3.45/5-st, mis ei ole väga hea tulemus, kuid tasub arvestada, et kõige madalamaid hindeid on siiski vaid mõni üksik. Kokku vastasid küsitlusele 87 kursusel osalejat. Punktiskaalal 1-5 hindasid rakenduse kasulikkust ka üliõpilased, kes ise rakendust ei paigaldanud ega lahendanud ka bonusülesannet. See tulemus seab kahtluse alla rakenduse kasulikkuse juhul, kui seda vaid praktikumis õppuritele demonstreerida, ilma, et nad ise rakendust üles seaksid ega oma isikliku andmebaasiga seaksid. Oluline on siiski lasta õppijatel ise rakendust üles seada. Üliõpilaste vastuste jagunemist võib näha ka joonisel 9.

Kuiõrd aitas viimase praktikumis näidatud Male andmebaasi rakendus (ka Moodles oli selle aadress <https://male-ten.vercel.app/>) mõista, kuidas andmebaase igapäevaelus kasutatakse?

 Copy chart

87 responses



Joonis 9. Üliõpilaste vastused küsimusele: “Kuiõrd aitas viimases praktikumis näidatud Male andmebaasi rakendus mõista, kuidas andmebaase igapäevaelus kasutatakse?”

Boonusülesannet lahendas kokku 11 üliõpilast, kellest 7 andsid ka tagasisidet ülesande kohta. Tagasiside boonusülesandele oli õppurite poolt pigem negatiivne, suurim puudujääk on üliõpilaste arvates ülesande juhend, kus oleks võinud olla rohkem detaile. Probleemid tekkisid õppuritel just rakenduse algse seadistamise ja oma andmebaasi ühendamise. Siin tuleb välja juhendite koostamise keerukus, sest kõiki probleeme on raske ette näha ning üliõpilaste tase on erinev. Probleemid tekkisid siis, kui etteantud käsud kohe ei toimunud, vaid oli vaja veel midagi teha, näiteks lubada rakendus läbi tulemüüri. Autor võttis tagasisidet kuulda, ning parandas ülesande juhendit, parandades sõnastust ning lisades juurde detaile, kuidas rakendust käivitada.

Üliõpilased hindasid ülesannet ka liiga raskeks, sest enamus boonusülesande lahendajaid ei olnud varem tegelenud veebirakenduste arendamisega. Samuti toodi välja, et ülesanne oleks olnud kergem, kui rakenduse kohta oleks olnud põhjalikum juhend. Kursusel “Andmebaasid” osalevad lisaks informaatika õppekava üliõpilastele ka näiteks matemaatilise statistika üliõpilased, kelle jaoks oli ülesanne palju raskem. Üks matemaatilise statistika õppekava üliõpilane väitis, et tal puudub igasugune varasem kogemus sarnase koodiprojektiga töötamisel ning ülesanne oli tema jaoks pea ületamatu raskusega. Kursusel osalevate õppurite tasemete erinevuse tõttu ongi üheselt mõistetava juhendi ja paraja raskusastmega ülesande koostamine väga raske. Kui lisada detaile liiga palju, siis võib ülesanne olla jällegi liiga piirav

ning mõjuda negatiivselt üliõpilaste õpimotivatsioonile. Töö autor parandas ja täiendas siiski veel ülesande juhendit, et muuta ülesande lahendamist lihtsamaks.

Kokkuvõtvalt võib rakenduse kohta öelda, et kursuse “Andmebaasid” õppejõududele rakendus meeldis ning arvati, et sellest võiks kasu olla. Kursusel osalevatele õppuritele jäi püstitatud boonusülesanne aga natuke keeruliseks ning juhis selle kohta liiga lühikeseks.

Kokkuvõte

Bakalaureusetöö eesmärk oli luua näitlikustav veebirakendus, mida saaks kasutada Tartu Ülikooli kursusel “Andmebaasid” õppevahendina, loomaks üliõpilastele selgemat pilti andmebaaside olulisusest tänapäevases maailmas. Selle täitmiseks arendas autor 2025. aasta kevadsemestriks male andmebaasil põhineva veebirakenduse ning boonusülesande, mis aine praktikumijuhendajate poolt kasutusele võeti.

Rakenduse arendamisel on kasutatud Vue.js raamistikku eesrakenduse ning Node.js serverit koos express.js raamistikuga tagarakenduse loomiseks. Rakenduse kasutajaliidese loomiseks on kasutatud ka Vue.js jaoks loodud komponentide teeki Vuetify. Rakendus tuleb üliõpilasetel enda arvutisse paigaldada ja enda andmabaasiga ühendada, et lahendada boonusülesannet. Rakendus jookseb ka konteinerisüsteemis Docker.

Tagasisidet rakenduse kohta koguti küsimustike abil nii kursuse “Andmebaasid” õppejõududelt kui ka kursuse üliõpilastelt. Õppejõudude tagasiside põhjal oli tegemist hea ja kasuliku rakendusega, mis täidab oma eesmärgi. Hinnati, et rakenduse nägemine ja ülesande lahendamine aitab luua üliõpilastele konteksti, kuidas andmebaase rakenduse osana kasutatakse ning üldisema pildi nägemine aitab tõsta õppijate õpimotivatsiooni. Üliõpilaste tagasisidest tuli välja, et rakenduse ülesseadmise õpetus ja boonusülesande püstitus oleks võinud olla detailsemalt ja põhjalikumalt lahti seletatud. Üldpildis mõjus rakendus siiski ka üliõpilastele positiivselt. Edasiarendusena on võimalik rakenduses ka paremini välja tuua vead seoses andmebaasiga, mis eeldaks andmebaasi vigade analüüsimist ja kasutajale sobilikule vormile ümber teisendamist.

Tagasiside põhjal võib väita, et selles bakalaureusetöös kasutatud rakendust saab kasutada andmebaaside õpetamisel õppijatele selgema konteksti loomiseks ja seeläbi õppimise lihtsustamiseks. Tasub siiski arvestada, et tagasisideküsitlusele vastasid alla poolte üliõpilastest ning pooled kursuse “Andmebaasid” õppejõududest.

Viidatud kirjandus

- [1] ÕIS II, <https://ois2.ut.ee/#/courses/LTAT.03.004/details> (27.03.2025).
- [2] Villa AH. Reinvigorating a database systems course using web applications. *J Comput Sci Coll* 2013; 28: 201–202.
- [3] School of Computer Science, Kean University, Union, New Jersey, USA, Huang C-Y. Learning Database through Developing Database Web Applications. *Int J Inf Educ Technol* 2019; 9: 241–249.
- [4] OpenAI (2022). ChatGPT (16. aprilli versioon): <https://chat.openai.com>
- [5] AKIT, <https://akit.cyber.ee/> (09.05.2025).
- [6] Elmasri R, Navathe S. *Fundamentals of database systems*. Seventh edition. Boston Munich: Pearson, 2016.
- [7] Coronel C, Morris S. *Database systems: design, implementation, and management*. Thirteenth edition. Mason, OH: Cengage Learning US, 2018.
- [8] Connolly TM, Begg CE. *Database systems: a practical approach to design, implementation, and management*. 6th edition, global edition. Boston Columbus Indianapolis: Pearson, 2015.
- [9] Qiu D, Li B, Su Z. An empirical analysis of the co-evolution of schema and code in database applications. In: *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. Saint Petersburg Russia: ACM, pp. 125–135.
- [10] Keller JM. Development and use of the ARCS model of instructional design. *J Instr Dev* 1987; 10: 2–10.
- [11] Fowler M. *Patterns of enterprise application architecture*. Boston: Addison-Wesley, 2003.
- [12] Richards M, Ford N. *Fundamentals of software architecture: an engineering approach*. First edition. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, 2020.
- [13] Luik P. Andmebaasid 1. loeng, https://moodle.ut.ee/pluginfile.php/2021259/mod_resource/content/8/L01_Andmebaasid_sissejuhatus_2024_v.pdf (27.03.2025).
- [14] Hule K, Ranawat R. Analysis of Different ORM Tools for Data Access Object Tier Generation: A Brief Study. *Int J Membr Sci Technol* 2023; 10: 1277–1291.
- [15] Colley D, Stanier C, Asaduzzaman M. The Impact of Object-Relational Mapping Frameworks on Relational Query Performance. In: *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. Southend, United Kingdom: IEEE, pp. 47–52.
- [16] Malhotra R. *Rapid Java Persistence and Microservices: Persistence Made Easy Using*

Java EE8, JPA and Spring. Berkeley, CA: Apress. Epub ahead of print 2019. DOI: 10.1007/978-1-4842-4476-0.

- [17] Elo rating system. *Wikipedia*, https://en.wikipedia.org/w/index.php?title=Elo_rating_system&oldid=1282954176 (2025, 08.05.2025).
- [18] Andmebaasid (LTAT.03.004) | TÜ Moodle, <https://moodle.ut.ee/course/view.php?id=11740> (12.05.2025).
- [19] kare22/ut-db-orm. *GitHub*, <https://github.com/kare22/ut-db-orm> (07.05.2025).
- [20] Sommerville I. *Software engineering*. 9th ed. Boston: Pearson, 2011.
- [21] Introduction to the server side - Learn web development | MDN, https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/First_steps/Introduction (2025, 07.05.2025).
- [22] ÕIS II, <https://ois2.ut.ee/#/courses/LTAT.05.004/version/d89279f4-f6c3-669a-7d02-0e42444b150c/details> (07.05.2025).
- [23] Vue.js, <https://vuejs.org/> (09.05.2025).
- [24] Technology | 2024 Stack Overflow Developer Survey, <https://survey.stackoverflow.co/2024/technology> (14.04.2025).
- [25] Vite. *vitejs*, <https://vite.dev> (09.05.2025).
- [26] webpack. *webpack*, <https://webpack.js.org/> (09.05.2025).
- [27] Vuetify — A Vue Component Framework. *Vuetify*, <https://vuetifyjs.com/en/> (07.05.2025).
- [28] unplugin-vue-router. *npm*, <https://www.npmjs.com/package/unplugin-vue-router> (2023, 14.04.2025).
- [29] Axios API | Axios Docs, https://axios-http.com/docs/api_intro (14.04.2025).
- [30] Express - Node.js web application framework, <https://expressjs.com/> (09.04.2025).
- [31] node-postgres, <https://node-postgres.com/> (14.04.2025).
- [32] Run multiple processes in a container. *Docker Documentation*, https://docs.docker.com/engine/containers/multi-service_container/ (09.05.2025).
- [33] Docker Compose. *Docker Documentation*, <https://docs.docker.com/compose/> (09.05.2025).
- [34] Vercel: Build and deploy the best web experiences with the Frontend Cloud. *Vercel*, <https://vercel.com/home> (09.05.2025).
- [35] Cloud Application Platform. *Render*, <https://render.com> (09.05.2025).

- [36] Supabase | The Open Source Firebase Alternative. *Supabase*, <https://supabase.com/> (09.05.2025).
- [37] Critical rendering path - Web performance | MDN, https://developer.mozilla.org/en-US/docs/Web/Performance/Guides/Critical_rendering_path (2025, 09.05.2025).
- [38] Sangarsu RR. Advancing Web Development with Single Page Applications (SPAs). *Journal of Scientific and Engineering Research*, 2019, 6(3); 284-288
- [39] Kothapalli M. Performance Analysis of Single Page Applications. *Int J Sci Res IJSR* 2022; 11: 1631–1635.

Lisad

1. Rakenduse lähtekood

Käesoleva bakalaureusetöö käigus valminud rakenduse lähtekood on saadaval GitHubi repositooriumis aadressil: <https://github.com/doktorjohan/male>

2. Küsimustik praktikumijuhendajatele

Rakendus aitab luua üliõpilase jaoks paremat ülevaadet, kuidas andmebaase igapäevaelus kasutatakse *

	1	2	3	4	5	
ei loo ülevaadet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	loob hea ülevaate

Usun, et boonusülesande lahendamine aitab üliõpilasel mõista andmebaaside olulisust *

	1	2	3	4	5	
ei aita mõista	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	aitab väga hästi mõista

Usun, et oma andmebaasi rakenduse osana nägemine parandab õppijate õppemotivatsiooni *

	1	2	3	4	5	
ei paranda	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	parandab väga hästi

Kokkuvõttes on rakendus üliõpilase jaoks kasulik *

	1	2	3	4	5	
pole üldse kasulik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	väga kasulik

Mis on rakenduses hästi teostatud? *

Pikk vastuse tekst

Mida tuleks rakenduses veel lihvida? *

Pikk vastuse tekst

3. Küsimustik üliõpilastele

Kuivõrd aitas viimases praktikumis näidatud Male andmebaasi rakendus (ka Moodles oli selle aadress <https://male-ten.vercel.app/>) mõista, kuidas andmebaase igapäevaelus kasutatakse?

1

2

3

4

5

Kui lahendasite boonusülesande, milles oli vaja rakendust täiendada, palun kirjutage, mis selle ülesande juurdes teile meeldis või mida annaks paremaks muuta?

Pikk vastuse tekst

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Johan Hollak

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Ainesse “Andmebaasid” näidise rakenduse loomine male andmebaasi põhjal

mille juhendaja on Piret Luik

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;

2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;

3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;

4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Johan Hollak

14.05.2025