

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Daniil Iljin 213299IADB

# **Müügiplatvorm veebirakendusena rõivaste sekundaarturu toetamiseks**

Bakalaureusetöö

Juhendaja: Lembit Viilup  
PhD

Tallinn 2025

## **Autorideklaratsioon**

Kinnitan, et olen 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.

Autor: Daniil Iljin

03.01.2025

## **Annotatsioon**

Kasutatud rõivaste utiliseerimise ja taaskasutamise probleem on muutunud üha aktuaalsemaks. Paljud riided, mis on veel kvaliteetsed ja nõutud, jäävad kasutamata, kuna müüjate ja ostjate kokkupanek on keeruline. Selle tagajärjel visatakse riided sageli minema või antakse taaskasutuskeskustesse.

Töö eesmärgiks on luua veebiplatvormi prototüüp, mis võimaldab lihtsat ja mugavat riiete müüki ja ostu, edendades jätkusuutlikku moe tarbimist ja toetades sekundaarse rõivaturu arengut.

Töös alustatakse teema aktuaalsuse ja olemasolevate lahenduste analüüsist, mis omakorda määrab platvormi arenduse eesmärgid ja ülesanded. Edasi, põhinedes analüüsil, valitakse sobivad tehnoloogiad ja arhitektuur, mis tagavad rakenduse efektiivsuse, skaleeritavuse ja turvalisuse. Need tehnilised valikud annavad aluse prototüübi loomisele, kus arendatakse edasi andmebaasi, tagarakenduse ja eesrakenduse funktsionaalsus, et pakkuda lõppkasutajale sujuvat ja turvalist kasutajakogemust.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 31 leheküljel, 5 peatükki, 9 joonist, 5 tabelit.

## **Abstract**

### **Sales Platform as a Web Application for Supporting the Secondary Clothing Market**

The problem of used clothing disposal and recycling has become increasingly important. Many clothes that are still in good condition and in demand are left unused because it is difficult to connect buyers and sellers. As a result, clothes are often thrown away or sent to recycling centers.

The goal of this project is to create a web platform prototype that allows for easy and convenient buying and selling of clothes, promoting sustainable fashion consumption and supporting the development of the second-hand clothing market.

The work starts with analyzing the relevance of the topic and existing solutions, which then defines the platform's development goals and tasks. Based on this analysis, the appropriate technologies and architecture are chosen to ensure the application's efficiency, scalability, and security. These technical decisions form the foundation for developing the prototype, where the database, backend, and frontend functionalities are further developed to provide a smooth and secure user experience.

The thesis is in Estonian and contains 31 pages of text, 5 chapters, 9 figures, 5 tables.

## Lühendite ja mõistete sõnastik

ACID	ACID ( <i>Atomicity, Consistency, Isolation, Durability</i> ) on andmebaasi transaktsioonide omadused, mis tagavad, et kõik andmed on järjepidevad ja usaldusväärsed
API	API ( <i>Application Programming Interface</i> ) on liides, mis võimaldab erinevate tarkvarasüsteemide omavahelist suhtlust ja andmete vahetamist
AWS S3	AWS S3 ( <i>Amazon Web Services Simple Storage Service</i> ) on pilvepõhine andmesalvestusteenus, mis võimaldab suurte andmemahtude salvestamist ja haldamist
AWS SDK	AWS SDK ( <i>Amazon Web Services Software Development Kit</i> ) on komplekt tööriistu ja teeke, mis aitavad luua rakendusi, mis suhtlevad AWS teenustega, nagu S3
CRUD	CRUD ( <i>Create, Read, Update, Delete</i> ) on andmebaasi toimingute põhilised kategooriad, mis võimaldavad andmete loomist, lugemist, värskendamist ja kustutamist
DTO	DTO ( <i>Data Transfer Object</i> ) on objekt, mida kasutatakse andmete edastamiseks süsteemis või teenuste vahel
HTTP	HTTP ( <i>Hypertext Transfer Protocol</i> ) on protokoll, mida kasutatakse veebisõnumite edastamiseks
JWT	JWT ( <i>JSON Web Token</i> ) on avatud standard, mida kasutatakse turvaliseks andmeedastuseks veebirakendustes
JPA	JPA ( <i>Java Persistence API</i> ) on Java raamistiku osa, mis lihtsustab andmebaasi suhtlemist objektorienteeritud mudelite abil
JPQL	JPQL ( <i>Java Persistence Query Language</i> ) on keelepäring, mis võimaldab päringuid teha JPA objektide suhtes, päringud on sarnased SQL-le
MVP	MVP ( <i>Minimum Viable Product</i> ) on tarkvara arenduses mõiste, mis viitab toote lihtsaimale versioonile, portotüübile.
NoSQL	NoSQL on mitte-relatsiooniliste andmebaaside üldtermin, mis pakub paindlikkust suurte, struktureerimata andmete salvestamiseks
ORM	ORM ( <i>Object-Relational Mapping</i> ) on tehnika, mis ühendab objektorienteeritud programmeerimise ja relatsiooniliste

andmebaaside, võimaldades automaatselt andmete salvestamist ja päringute tegemist

REST API

REST API (*Representational State Transfer*) on teenuste arhitektuur, mis võimaldab veebirakenduste ja süsteemide vahel andmevahetust HTTP protokolliga kaudu.

SQL

SQL (*Structured Query Language*) on keel, mida kasutatakse relatsiooniliste andmebaaside haldamiseks ja päringute tegemiseks

UI

UI (*User Interface*) on kasutajaliides

URL

URL (*Uniform Resource Locator*) on aadressi formaat, mida kasutatakse veebis asukohtade (ressursside) määratlemiseks ja neile juurdepääsu saamiseks

## Sisukord

1 Sissejuhatus .....	11
2 Ülesande püstitus.....	12
2.1 Aktuaalsus .....	12
2.2 Olemasolevad lahendused .....	14
2.2.1 Sotsiaalmeedia (Instagram) .....	14
2.2.2 Müügiplatvormid.....	15
2.3 Eesmärk .....	17
2.4 Skoop.....	17
3 Loodava rakenduse analüüs.....	18
3.1 Nõuete määramine.....	18
3.2 Veebirakenduse arhitektuur.....	21
3.3 Tehnoloogiliste lahenduste valik.....	22
3.3.1 Andmebaasi tehnoloogiate valik .....	23
3.3.2 Tagarakenduse tehnoloogiate valik .....	26
3.3.3 Eesrakenduse tehnoloogiate valik .....	27
4 Veebirakenduse arendus.....	28
4.1 Andmebaasi projekteerimine ja loomine.....	28
4.2 Tagarakenduse arendus.....	31
4.2.1 Tagarakenduse ülesehitus .....	32
4.2.2 Andmebaaside haldamine Spring JPA abil .....	34
4.2.3 Rakenduse turvalisuse tagamine Spring Security abil.....	34
4.2.4 AWS S3 piltide üles- ja allalaadimiseks.....	35
4.3 Kasutajaliidese loomine.....	35
4.3.1 Eesrakenduse struktuuri ülevaade .....	35
4.3.2 Loodud kasutajaliidese ja selle funktsionaalsuse ülevaade .....	38
4.4 Prototüübi testimine.....	41
4.5 Edasise arendamise võimalused .....	42
5 Kokkuvõte .....	43
Kasutatud kirjandus .....	44

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	46
Lisa 2 – rakenduse repositoorium.....	47



## Jooniste loetelu

Joonis 1. Sekundaarse turu suuruse ja kasvu hinnangud [1] .....	12
Joonis 2. DB-Engines andmebaasisüsteemide populaarsuse trend (2014-2024) .....	24
Joonis 3. Rakenduse andmemudel.....	29
Joonis 4. Tagarakenduse projekti struktuur.....	32
Joonis 5. Eesrakenduse projekti struktuur.....	36
Joonis 6. Rakenduse avaleht.....	38
Joonis 7. Autoriseeritud kasutaja avaleht .....	39
Joonis 8. Toote detailvaade .....	40
Joonis 9. Toote lisamise ja muutumise vaade.....	40

## **Tabelite loetelu**

Tabel 1. Mittefunktsionaalsed nõuded.....	19
Tabel 2. Funktsionaalsed nõuded .....	20
Tabel 3. Arenduse prioriteetide tabel .....	21
Tabel 4. Andmebaasi tüübide võrdlus. ....	23
Tabel 5. Andmebaaside tehnoloogiate võrdlus.....	25

## 1 Sissejuhatus

Üheks oluliseks globaalseks probleemiks on kujunenud kasutatud toodete utiliseerimine ja taaskasutamine. Probleem on selles, et kui mõningate tarbekaupade puhul on utiliseerimine lihtsamini realiseeritav, siis kasutatud rõivaste puhul on see suhteliselt keeruline. Samas on paljud rõivatööstuse tooted, mis enam kasutamist ei leia, igati kvaliteetsed ja oleksid sekundaarturul vägagi nõutud kaup. Kuna mainitu valdkonnas on müüjaid ja ostjaid suhteliselt palju, siis on osutunud keeruliseks ülesandeks osapoolte kokku viimine ja rõivaste müügi asemel visatakse need kas minema või antakse taaskasutuskeskustesse.

Seetõttu käesoleva töö eesmärgiks on luua spetsiaalne veebiplatvorm riiete müügiks, mis oleks mõeldud inimestele, kes tunnevad huvi jätkusuutliku moe ja soodsate riiete vastu. Platvorm peaks pakkuma lihtsat viisi, kuidas toetada ressurside säästlikku ja korduvkasutust ning samal ajal aidata kaasa rõivaste sekundaarturu arengule.

Sellise täielikult valmis platvormi loomine on üsna mahukas töö, mistõttu on käesoleva töö raames kavandatud luua rakenduse prototüüp või MVP. Kavandatud prototüüpi plaanitakse testida väikese testkasutajate grupi peal. Saadud tagasiside ja tulemuste põhjal on võimalik planeerida projekti edasist arendamist.

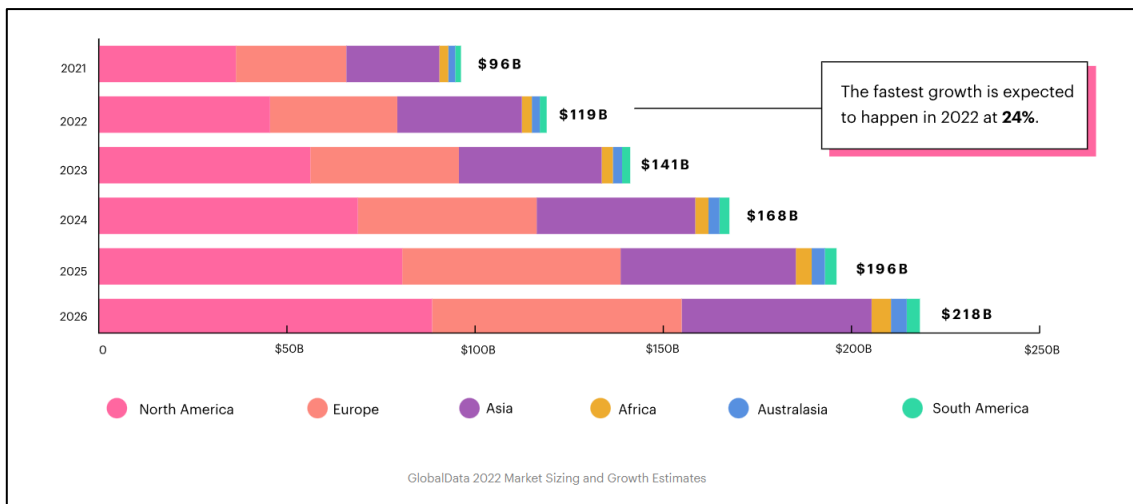
Käesoleva lõputöö kirjaliku osa struktuur on järgmine: esimeses osas käsitletakse töö aktuaalsust ja motivatsiooni; teises osas on fookuses loodavale rakendusele esitatud nõuded, rakenduse arhitektuur ja metoodika; kolmandas osas antakse ülevaade rakenduse arendamise protsessist; neljandas osas analüüsitakse loodud rakendust, saadud tulemusi ja hinnatakse teostatud tööd.

## 2 Ülesande püstitus

Käesolev peatükk annab põhjalikuma ülevaate sekundaarse turu arendamise aktuaalsusest ja käsitletavast probleemist. Samuti analüüsitakse olemasolevaid ja kasutatavaid lahendusi ning nende võimalusi. Lõpus esitatakse planeeritava töö skoop ja eesmärk.

### 2.1 Aktuaalsus

Kaasaegsed keskkonna- ja majandusväljakutsed sunnivad ümber mõtlema tarbimis- ja ressursside ümberjaotamise viise. Viimastel aastatel on sekundaarne turg saanud jätkusuutliku tarbimise lahutamatuks osaks, pakkudes tõhusaid lahendusi jäätmete vähendamiseks, loodusvarade koormuse vähendamiseks ja erinevate toodete majandusliku kättesaadavuse parandamiseks. Joonisel 1 on kujutatud sekundaarse turu laienemise trendi eri maailma piirkondades [1].



Joonis 1. Sekundaarse turu suuruse ja kasvu hinnangud [1]

Praegu on sekundarturu areng laialdaselt tingitud paljusidest teguritest, mille seas on keskse tähtsusega majanduskriisid ja elukalliduse kasv. Kriisiolukordades ja rahalise ebastabiilsuse tingimustes otsivad tarbijad odavamaid alternatiive, mis aitavad kulusid vähendada. Kohalikud taaskasutuspoed või kauplemisplatvormid nagu okidoki.ee ja osta.ee saavad pakkuda kvaliteetseid tooteid mõistlike hindadega, muutes need ostjate jaoks atraktiivseks. Selliste toodete hulka kuulub ka korduvkasutatud riietus. [1] [2]

Kasutatud riiete populaarsus on samuti suurenenud noorte ja sotsiaalmeedia toel. Kaasaegne moe mitmekesisus ja aktuaalsed suundumused loovad võimalusi leidmiseks stiilseid ja unikaalseid esemeid taaskasutuspooldest, võimaldades tarbijatel mitte kulutada raha kallitele uutele rõivastele. Samuti toetavad noored aktiivselt jätkusuutlikku tarbimist, kasutades taaskasutuspooldi eneseväljendamise ja unikaalse stiili kujundamise viisina. See trend näitab, et kasutatud esemete ostmise on muutumas normaalsuseks, luues uusi sotsiaalseid norme. [3] [4]

Lisaks on kasvav keskkonnateadlikkus veel üks stiimul üleminekuks sekundaarsele turule. Inimesed hakkavad mõistma kiirmoe negatiivseid tagajärgi, sealhulgas keskkonnasaastet, märkimisväärseid süsinikuheiteid ning liigset pakendite kasutamist. Sekundaarne turg riiete osas võimaldab vähendada jäätmeid ja pikendada asjade eluiga, mis on eriti atraktiivne tarbijatele, kes soovivad vähendada oma negatiivset mõju keskkonnale. [5]

Lõpuks on internetiteenuste ja platvormide areng muutnud ka kasutatud esemete ostmise protsessi mugavamaks. Veebiteenused, nagu okidoki.ee ja osta.ee, pakuvad laia tootevalikut, mis võimaldab ostjatel mugavalt leida unikaalseid esemeid ja võrrelda hindu. See tõmbab järjest rohkem inimesi sekundarturule, muutes selle olulise osaks kaasaegse tarbija elust.

Siin loetletud põhjused sekundaarse turu arenguks rõhutavad mugavate ja kättesaadavate lahenduste loomise olulisust taaskasutatud toodete ja rõivaste müügiks ja ostuks. Olemasolevad platvormid ei vasta aga alati kasutajate ootustele, mis nõuab uute ja tõhusamate lähenemisviiside leidmist.

## 2.2 Olemasolevad lahendused

Praegu ilmub pidevalt uusi veebiteenuseid, mille hulka kuuluvad erinevad veebikaubandusplatvormid, ning kasutajatele pakutakse arvukalt rakendusi ja veebiresursse kasutatud kaupade ostmiseks ja müümiseks. Kuna selliseid platvorme on väga palju, käsitletakse käesolevas töös ainult neid rakendusi, mida saab kasutada taaskasutatavate riide kauplemiseks. Analüüs keskendub Eestis kohalikuks kasutamiseks mõeldud lahenduste tugevustele ja puudustele. Analüüsis on lisatud ka sellised lahendused nagu Instagram ja Facebook Marketplace, kuna neid saab käsitleda kohaliku kasutuse kontekstis, olles populaarsed platvormid kasutatud riide müügiks.

### 2.2.1 Sotsiaalmeedia (Instagram)

Sotsiaalmeediat saab kasutada kaubanduslikel eesmärkidel, pakkudes kasutajatele lihtsat viisi oma toodete esitlemiseks ja müümiseks. Üheks heaks näiteks on Instagram, mis oma pildipõhise formaadi ja suure kasutajabaasi tõttu on saanud äärmiselt populaarseks müügikanaliks. Instagrami kasutajate arv ulatub üle 1 miljardi, ja neist 70% otsib regulaarselt uusi tooteid platvormi kaudu. Peaaegu 130 miljonit kasutajat klikib igakuiselt ostulinkidele, mis näitab Instagrami funktsioonide tõhusust ja kõrget kasutajate kaasatust. Tänu sellele saab Instagram pakkuda kiiret ja loomingulist suhtlemist laia vaatajaskonnaga, mis omakorda aitab kaasa müügi suurendamisele. [6]

Siiski on mitmeid piiranguid. Sisu nähtavus sõltub suuresti algoritmidest, mis võivad piirata ulatust ilma tasulise reklaamita. Reklaamikampaaniad on sageli kulukad ning platvormi puudujääk on ka integreeritud makselahenduste ja tellimuste halduse süsteemi puudumine. Seega muutub müügi protsessi juhtimine keeruliseks ja vajab tihti kolmandate osapoolte tööriistu.

Kokkuvõttes, kuigi Instagram võib tõhusalt kliente ligi meelitada, ei paku see spetsialiseeritud kaubandusplatvormide paindlikkust ega funktsionaalsust, mis teeb selle täieõigusliku müügikanalina kasutamise keeruliseks. [7]

### **2.2.2 Müügiplatvormid**

Tänapäeval on olemas palju erinevaid müügiplatvorme, mis võimaldavad kasutajatel müüa ja osta nii uusi kui ka kasutatud esemeid. Käesoleva töö kontekstis käsitletakse kolme kõige populaarsemat müügiplatvormi, mida Eestis aktiivselt asjade müümiseks kasutatakse. [8]

#### **Facebook Marketplace**

Facebook Marketplace, tänu oma integreerimisele Facebooki platvormiga, pakub kasutajatele mitmeid olulisi eeliseid. Üks silmapaistvamaid plusse on mugav suhtlemine läbi Facebook Messenger'i, mis lihtsustab tehingutingimuste arutamist ja vähendab vajadust kasutada kolmanda osapoolse rakendusi vestluseks. Platvorm pakub ka võimalust kohalikuks otsinguks, mis võimaldab kasutajatel filtreerida kuulutusi oma linna või piirkonna järgi. See on eriti kasulik kasutatud kaupade tehingute puhul, kuna ostjad saavad vältida täiendavaid kohaletoometamiskulusid.

Siiski on Facebook Marketplace'il ka oma puudused. Üks tõsisemaid probleeme on tehingute kaitsmiseks vajalike sisseehitatud tööriistade puudumine. Kasutajad kannavad täielikku vastutust iga tehingu turvalisuse eest, mis muudab platvormi haavatavaks petturite suhtes. Lisaks, erinevalt spetsialiseeritud kaubanduse platvormidest, ei paku Facebook Marketplace võimalust müüjate hindamiseks, mis muudab nende maine ja usaldusvääruse kontrollimise keeruliseks. See võib olla probleemiks kasutajatele, kes otsivad turvalisemaid ja usaldusväärsemaid kanaleid ostmiseks ja müügiks. [9]

#### **Osta.ee**

Osta.ee on üks Eesti juhtivaid e-kaubanduse platvorme, mis pakub laia valikut ostu- ja müügivõimalusi erinevate toodete ja teenuste jaoks. Selle peamiseks tugevuseks on turvalised ja usaldusväärased lahendused, sealhulgas maksete garantii- ja pettusevastased süsteemid, mis aitavad vähendada tehingutega seotud riske. Samuti on kasutusel tagasiside- ja reitingusüsteem, mis suurendab usaldusväärust, aidates kasutajatel leida kvaliteetseid müüjaid ja vältida madala kvaliteediga tooteid.

Platvorm on suunatud nii eraisikutele kui ka väikestele ja keskmise suurusega ettevõtetele, pakkudes lihtsat ja paindlikku lahendust toodete müümiseks. Osta.ee kasutajaliides on loodud kasutajasõbralikuks, võimaldades hõlpsalt lisada ja hallata tooteid erinevates kategooriates. Lisaks pakub platvorm reklaamivõimalusi, et suurendada toodete ja teenuste nähtavust ning parandada müügitulemusi. See teeb Osta.ee-st usaldusväärse ja tõhusa tööriista nii eraisikute kui ettevõtete jaoks. [10]

Samas on Osta.ee-l ka puudused. Üheks suurimaks probleemiks on turvalisus, eriti isiklikel kohtumistel tehtavate tehingute puhul. Kuigi platvorm pakub teatud kaitsemeetmeid, ei pruugi need alati pettuste eest täielikult kaitsta. Näiteks on olnud juhtumeid, kus müüja saadab ostjale tühja paki või toote, mis ei vasta kirjeldusele. [11]

### **Okidoki.ee**

Okidoki.ee on Eesti turul tegutsev kuulutuste platvorm, millel on Osta.ee-ga üsna sarnane funktsionaalsus, võimaldades kasutajatel oma tooteid ja teenuseid mugavalt müügiks pakkuda. Platvormi lihtne ja kasutajasõbralik keskkond muudab kuulutuste lisamise kiireks ja kergesti ligipääsetavaks, olles eriti atraktiivne neile, kes soovivad oma pakkumised suunata kohalikele ostjatele.

Samas on Okidoki.ee-l mõned olulised piirangud. Erinevalt Osta.ee-st ei paku platvorm tehingute kaitseüsteemi, mis muudab kasutajad potentsiaalsete riskide suhtes vähem kaitstuks. Samuti puudub Okidoki.ee-l oksjonisüsteem, mis on Osta.ee üks populaarsemaid funktsioone ja pakub paindlikkust ostjatele pakkumiste tegemisel ning müüjatele võimalust toodete müügihinda konkurentsi kaudu tõsta. Nende piirangute tõttu ei pruugi Okidoki.ee olla ideaalne valik neile, kes eelistavad turvalisemat ja mitmekülgsemat müügikeskkonda.



## **2.3 Eesmärk**

Selle lõputöö eesmärk on välja töötada spetsialiseeritud veebiplatvormi prototüüp taaskasutatud rõivaste müügiks. Projekti eriline omadus seisneb platvormi spetsialiseerumises taaskasutatud rõivaste kauplemisele, eesmärgiga muuta see protsess mugavamaks ja lihtsamaks, võttes arvesse rõivaste sekundaarse turu spetsiifilisi vajadusi.

## **2.4 Skoop**

Täisfunktsionaalse rakenduse väljatöötamine, nagu eespool kirjeldatud, nõuab märkimisväärseid ajalisi ja rahalisi ressursse. Võttes arvesse lõputöö mahtu ning autori rahalisi ja oskuste piiranguid, ei ole käesoleva projekti raames võimalik valmis arendada täisversiooni platvormist.

Platvormi minimaalne funktsionaalsus sisaldab kasutajate registreerimist, profiilide loomist, kuulutuste lisamist koos fotode ja kirjeldusega, otsingu- ja filtreerimissüsteemi tootekategooriate ja omaduste järgi (näiteks suurus, kaubamärk). Kuid oluline on märkida, et projektiversioon ei sisalda maksesüsteemi – makseprotsess täpsustatakse siis, kui ostja võtab müüjaga ühendust kuulutuses märgitud kontaktandmete kaudu.

Platvorm on loodud Eesti kasutatud rõivaste turu toetamiseks ja arendamiseks. Selleks planeeritakse rakenduse kasutajaliides esialgselt luua eesti keeles, et tagada kohalikele kasutajatele selge ja arusaadav kogemus. Prototüübi nõuete täpsem kirjeldus on esitatud peatükis „Loodava rakenduse analüüs“.

## 3 Loodava rakenduse analüüs

Käesolevas peatükis on esitatud nõuete analüüs ning arhitektuuri- ja tehnoloogiliste lahenduste põhjendamine, mis on vajalikud rakenduse arendamiseks. Analüüsi põhieesmärk on määratleda funktsionaalsed ja mittefunktsionaalsed nõuded, valida sobiv arhitektuurimudel ning põhjendada erinevate tehnoloogiate kasutamist.

Teostatud analüüsi põhjal tehakse strateegilised otsused, mis loovad tugeva aluse rakenduse arendamiseks, tagades selle vajaliku funktsionaalsuse, turvalisuse ja skaleeritavuse. Lisaks suunab analüüsi tulemus edasisi arendustegevusi, et rakendus vastaks kasutajate vajadustele ning aitaks tõhusalt saavutada seatud ärieesmärke.

### 3.1 Nõuete määramine

Selles peatükis kirjeldatakse loodava veebirakenduse nõudeid, mis jagunevad funktsionaalseteks ja mittefunktsionaalseteks. Nõuded määratlevad, millised funktsioonid ja omadused on vajalikud rakenduse tõhusaks tööks ja kasutajate ootuste täitmiseks.

Edasi kirjeldatud nõuded põhinevad autori kogemusel nii ostja kui ka müüja vaatenurgast, võttes arvesse nende peamisi vajadusi ja väljakutseid müügiplatvormide kasutamisel.

Selguse huvides on nõuded jaotatud kaheks tabeliks (tabel 1 ja tabel 2): mittefunktsionaalsed ja funktsionaalsed nõuded, millest igäüks kajastab rakenduse põhiomadusi.

Nõuete tabelid sisaldavad järgmisi veerge:

1. **Id** — nõude unikaalne identifikaator, mis võimaldab sellele dokumentatsioonis kergesti viidata.
2. **Nimetus** — nõude lühikirjeldus.
3. **Roll** — sihtrühm, kelle jaoks nõue on mõeldud (näiteks administraator, tavakasutaja), et arvestada erinevate kasutajagruppide vajadustega.

4. **Kirjeldus** — nõude täpsem selgitus, mis toob esile, mida tuleb realiseerida või tagada.
5. **Väärtus** — nõude olulisus rakenduse kasutaja jaoks (kõrge / madal) hinnatud autori kogemuse põhjal.
6. **Keerukus** — nõude realiseerimise eeldatav keerukus või maksumus (kõrge / madal), määratud autori varasema kogemuse järgi.

Funktsionaalsed nõuded kirjeldavad, milliseid võimalusi ja tegevusi rakendus peab pakkuma. Mittefunktsionaalsed nõuded seevastu keskenduvad rakenduse omadustele ja piirangutele, näiteks jõudlusele, turvalisusele ja skaleeritavusele, mis on vajalikud, et tagada rakenduse stabiilne ja usaldusväärne toimimine. Järgmistes tabelites algab funktsionaalsete nõute ID tähisega F ja mittefunktsionaalsete nõute ID tähisega NF.

Tabel 1. Mittefunktsionaalsed nõuded

Id	Nimetus	Kirjeldus	Väärtus	Keerukus
NF1	Adaptiivne disain	Veebirakendus kohandub erinevate seadmete ja ekraanisuurustega	Madal	Madal
NF2	Lihtne kasutatavus ja navigeerimine	Rakendus on kasutajasõbralik, intuitiivne ja võimaldab mugavat navigeerimist.	Kõrge	Madal
NF3	Andmete turvaline ligipääs	Tagatakse kasutajaandmete kaitse ning volitamata juurdepääsu ennetamine.	Kõrge	Kõrge

Tabel 2. Funktsionaalsed nõuded

ID	Nimetus	Roll	Kirjeldus	Väärtus	Keerukus
F1	Kasutaja registreerimine ja autoriseerimine	Ostja, Müüja	Võimaldab kasutajatel rakendusse registreeruda, sisse logida.	Kõrge	Madal
F2	Kuulutuste filtreerimine kategooriate ja brändide järgi	Ostja	Võimaldab kasutajatel filtreerida kuulutusi kategooriate ja brändide alusel, et hõlbustada toodete otsingut.	Kõrge	Kõrge
F3	Kuulutuste sorteerimine hinna järgi	Ostja	Annab kasutajatele võimaluse sorteerida kuulutusi hinna alusel kasvavas või kahanevas järjekorras.	Kõrge	Madal
F4	Kasutaja töölaua ja kuulutuste haldamine	Müüja	Pakub autoriseeritud kasutajatele töölaua, kus saab vaadata ja hallata enda loodud kuulutusi, sealhulgas neid muuta või kustutada.	Kõrge	Kõrge
F5	Lemmikute nimekiri	Ostja	Võimaldab kasutajatel ajutiselt salvestada huvipakkuvaid kuulutusi "Lemmikute" vaatesse, et neile kiiresti ligi pääseda.	Kõrge	Madal
F6	Kuulutuste loomine ja muutmine	Müüja	Võimaldab müüjatel luua uusi kuulutusi, muuta olemasolevaid ning lisada kirjeldavat teavet.	Kõrge	Kõrge
F7	Piltide üleslaadimine ja kuvamine	Müüja	Võimaldab kasutajatel kuulutustele pilte üles laadida ja neid koos kuulutuse infoga kuvada.	Kõrge	Kõrge
F9	Müüja kontaktandmete kuvamine	Ostja	Võimaldab ostjatel näha müüja kontaktandmeid, pakkudes lihtsustatud viisi toodete ostmiseks ja müümiseks.	Kõrge	Madal

Varem esitatud funktsionaalsete ja mittefunktsionaalsete nõuete tabelite põhjal on koostatud prioriteetide tabel 3, mis aitab määrata, millised ülesanded tuleb esimeses järjekorras täita ja milliseid saab edasi lükata. See arvestab iga nõude tähtsust ja teostamise keerukust, võimaldades arendusprotsessi tõhusamalt planeerida ning keskenduda kõige olulisematele ülesannetele.

Tabel 3. Arenduse prioriteetide tabel

	<b>Kõrge keerukus</b>	<b>Madal keerukus</b>
<b>Kõrge väärtus</b>	F2, F4, F6, F7, NF3	F1, F3, F5, F8, NF2
<b>Madal väärtus</b>		F6, NF1

### 3.2 Veebirakenduse arhitektuur

Mitme tasandi arhitektuur (või kihiline arhitektuur) on lähenemisviis, kus rakendus jagatakse mitmeks loogiliseks kihiks, millest igaüks täidab oma spetsiifilise ülesande. Tavaliselt eristatakse kolme põhikihti: esitluskiht (UI), äri loogika kiht ja andmekihi kiht

Mitme tasandi arhitektuuri eelised:

- Modulaarsus ja skaleeritavus: iga kihti saab muuta või skaleerida sõltumatult, mis muudab süsteemi paindlikuks ja hõlpsasti hallatavaks.
- Koodi taaskasutus: äri loogika ja andmekihid võivad olla kasutatavad mitmesugustes rakendustes, kui need on korralikult eraldatud.
- Turvalisus ja isoleeritus: iga kiht täidab oma rolli, minimeerides komponentidevahelist suhtlust. See suurendab rakenduse turvalisust, kuna ühe kihi muudatused ei mõjuta teisi.
- Hooldatavus: kihiliste struktuuride puhul on lihtsam vigu diagnoosida ja parandada, kuna iga kihi loogika on eraldi.

Mitme tasandi arhitektuuri puudused:

- Keerukus: mitme tasandi süsteemide projekteerimine võib olla keeruline, eriti juhul, kui kihtide arv suureneb.
- Tõhususe langus: mitmed kihid võivad vähendada jõudlust, kuna kihtidevahelised suhtlused suurenevad.
- Lisakulud: tasandite korraldamiseks ja toetamiseks on vaja rohkem ressursse, eriti suurtes rakendustes.

See lähenemine sobib ideaalselt suurte ja keerukate süsteemide jaoks, kus on oluline jagada loogika tasanditeks, mida saab arendada ja testida sõltumatult. Samuti võib see olla kasulik veebirakenduste puhul, kus funktsioonide selge jaotus on vajalik, näiteks turvalisuse, skaleeritavuse tagamiseks või süsteemi osade hõlpsaks uuendamiseks. [12]

Selle projekti puhul, kuigi monoliitne arhitektuur oleks samuti sobilik selle projekti jaoks, on mitme tasandi süsteem optimaalsem. Monoliitne lähenemine võib olla kasulik kiire prototüüpimise jaoks, kuid mitme tasandi arhitektuur pakub olulisi eeliseid skaleeritavuse ja pikaajalise toe osas. Lisaks lihtsustab mitme tasandi lähenemine rakenduse testimist ja hooldust, kuna iga kihti saab testida ja muuta ilma, et see tugevalt mõjutaks teisi süsteemi osi.

### **3.3 Tehnoloogiliste lahenduste valik**

Tarkvara loomisel on üks peamisi etappe tehnoloogiate valik, mida arendamisel kasutatakse. See valik määrab, kui tõhusalt lahendatakse püstitatud ülesandeid, samuti mõjutab see arenduse kiirust, jõudlust ja lõpptulemuse töökindlust. Käesolevas peatükis käsitletakse ja analüüsitakse põhjalikult rakenduse arendamiseks vajalike tehnoloogiate valikut, sealhulgas programmeerimiskeeli, raamistikke, andmebaase, testimistööriistu ja teisi komponente.

Õige tehnoloogiate valik mängib projekti edukas elluviimises otsustavat rolli. Kaasaegsete ja sobivate tööriistade kasutamine mitte ainult ei lihtsusta arendusprotsessi, vaid vähendab võimalikke kulusid ning kiirendab arendusprotsessi.

Arenduses plaanitakse arvestada ka isiklikku kogemust valitud tehnoloogiatega töötamisel. Tuttavate tööriistade kasutamine aitab vältida paljusid vigu ning vähendada aja raiskamist. Selline lähenemine võimaldab tehnoloogiate valikut teadvustatumalt käsitleda ning arvestada võimalike piirangute ja riskidega.

### 3.3.1 Andmebaasi tehnoloogiate valik

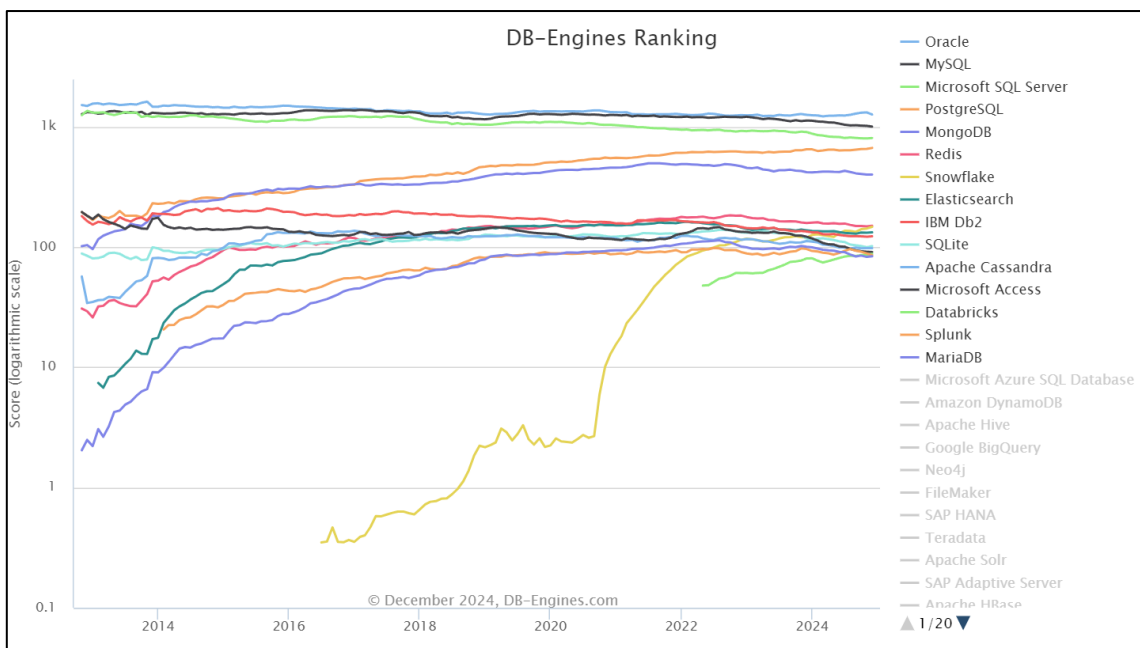
Kõigepealt tuleb määrata kasutatava andmebaasi tüüp. Peamised tüübid on SQL (relatsioonilised andmebaasid) ja NoSQL (mitterelatsioonilised andmebaasid). Mõlemal tüübil on oma eelised ja puudused, mis sobivad erinevatesse kasutusjuhtudesse. Alljärgnevas tabelis 4 on esitatud nende kahe tüübi põhivõrdlus (tabel põhineb TestGorilla blogi andmetel) [13]:

Tabel 4. Andmebaasi tüüpide võrdlus.

Omadus	SQL-andmebaasid	NoSQL-andmebaasid
<b>Skeem</b>	Fikseeritud, struktureeritud andmed	Paindlik, struktureerimata andmed
<b>Laiendatavus</b>	Vertikaalne (riistvara uuendus)	Horisontaalne (serverite lisamine)
<b>Tulemuslikkus</b>	Kiirem keerukates päringutes	Kiirem suurtel, struktureerimata andmetel
<b>ACID-vastavus</b>	Tugev, tagab järjepidevuse	Mõnel puudub täielik ACID-tugi
<b>Kulu</b>	Kõrgem, vertikaalse skaleeritavuse tõttu	Madalam, tänu horisontaalsele skaleerimisele
<b>Näited</b>	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, DynamoDB
<b>Sobivus</b>	Andmete terviklikkus (nt finantssüsteemid)	Suured andmed ja reaalajas rakendused

Kuna antud lõputöö projektis on vajalik teatud andmestruktuuri kasutamine, olenemata sellest, et tegemist on prototüübi loomisega, ning kuna töö autoril on rohkem kogemusi SQL-andmebaasidega, oleks parimaks valikuks SQL-andmebaas. [14]

Järgmisena tuleb valida konkreetne andmebaasitehnoloogia. Põhinedes platvormil DB-Engines esitatud andmetel joonisel 2, valis autor võrdlemiseks neli populaarset relatsiooniandmebaasi: PostgreSQL, Oracle, MySQL ja Microsoft SQL Server. Need andmebaaside haldussüsteemid (DBMS) asuvad edetabeli tippudes ja on laialdaselt kasutusel erinevates valdkondades, alates väikestest projektidest kuni suurte korporatiivsete lahendusteni. [15]



Joonis 2. DB-Engines andmebaasisüsteemide populaarsuse trend (2014-2024)



Allpool tabelis 5 on toodud ülevaade nendest andmebaasidest ja nende põhifunktsioonidest. Võrdlus on esitatud tabelis [16]:

Tabel 5. Andmebaaside tehnoloogiate võrdlus.

Kriteerium	MySQL	PostgreSQL	Microsoft SQL Server	Oracle Database
<b>Litsents</b>	Avatud lähtekoodiga	Avatud lähtekoodiga	Kommertsliitsents	Kommertsliitsents
<b>Jõudlus</b>	Kõrge lugemisoperatsioonide korral, piirangud kirjutamisel	Tasakaalustatud lugemise ja kirjutamise jõudlus	Kõrge jõudlus	Turu juhtiv jõudlus
<b>SQL standardite tugi</b>	Hea, kuid mitte täiuslik	Parim SQL standardite tugi	Hea	Laiendustega parim standardite tugi
<b>Laiendatavus</b>	Piiratud	Kõrge (laienduste tugi)	Hästi skaleeritav	Kõrge, sobib suurorganisatsioonidele
<b>Platvormide tugi</b>	Windows, Linux, MacOS	Windows, Linux, MacOS	Ainult Windows	Windows, Linux
<b>Maksumus</b>	Tasuta	Tasuta	Nõuab litsentsi	Kõrged litsentsikulud

Selle projekti jaoks plaanitakse kasutada PostgreSQL-i, kuna see on tasuta tehnoloogia, mis pakub kõrget SQL-standardite ühilduvust ja paindlikku laiendatavust. Lisaks on autoril PostgreSQL-iga kõige rohkem kogemusi, mis võimaldab prototüübi kiiret ja tõhusat elluviimist.

### **3.3.2 Tagarakenduse tehnoloogiate valik**

Serveri poole tehnoloogia valikul on oluline arvestada programmeerimiskeele populaarsuse, jõudluse ja eripäradega. Orient Software'i andmetel on enimkasutatavad serveripoolsed programmeerimiskeeled JavaScript (Node.js), Python, Java, PHP, Ruby, C# ja Go. Igal neist on omad tugevused ja nõrkused, sõltuvalt projekti nõudmistest.

#### **JavaScript (Node.js)**

JavaScript, kasutades Node.js platvormi, võimaldab kasutada ühte keelt nii serveri kui ka kliendipoolse arendamiseks. See muudab arenduse lihtsamaks ja ühtsemaks, eriti väiksemate meeskondade jaoks. Node.js on kõrge jõudlusega, kuna see toetab asünkroonset andmetöötlust. Kuid ressursimahukate arvutusülesannete puhul võib Node.js jääda alla mõnele teisele keelele.

#### **Python**

Python on tuntud oma lihtsuse ja ulatusliku teekide valiku poolest. Raamistikud nagu Django ja Flask võimaldavad kiiresti luua prototüüpe ja skaleeritavaid serverirakendusi. Python sobib eriti hästi andmetöötluse ja masinõppe projektidesse, kuid võib kõrge koormuse korral jääda vähem jõudluskõlblikuks kui näiteks Java.

#### **Java**

Java on üks populaarsemaid keeli ärirakenduste arendamiseks. Selle peamised eelised on kõrge jõudlus, skaleeritavus ja turvalisus. Raamistikud nagu Spring pakuvad tugevaid tööriistu serverirakenduste loomiseks. Java on sageli kasutusel suurtes süsteemides, kus on vaja kõrget töökindlust ja suurt kasutajamahu toetust.

#### **PHP**

PHP-d kasutatakse traditsiooniliselt veebilehtede arendamiseks. See on lihtne õppida ja sellel on lai ökosüsteem, sealhulgas populaarsed raamistikud nagu Laravel ja Symfony. PHP sobib hästi HTML-i genereerimise ja kasutajapäringute töötlemise ülesanneteks, kuid jääb jõudluse ja skaleeritavuse poolest teistele keelte alla.

## **Ruby**

Ruby koos Ruby on Rails raamistikuga on tuntud oma lihtsuse ja kiire arendusprotsessi poolest. See keel sobib hästi alustavatele ettevõtetele ja väiksematele projektidele. Siiski on Ruby jõudlus ja populaarsus viimastel aastatel langenud, mis piirab spetsialistide ja tööriistade valikut.

## **C#**

C# kasutatakse peamiselt .NET platvormil, mis teeb selle heaks valikuks Windowsi rakenduste jaoks. See sobib keerukate ärilahenduste ja veebiteenuste loomiseks. Kuid C# võib olla keerulisem õppida ja selle kasutamine sõltub suuresti Microsofti ökosüsteemist.

## **Go**

Go (või Golang), mille töötas välja Google, on mõeldud kõrge jõudluse ja skaleeritavate rakenduste loomiseks. Seda keelt iseloomustavad lihtsus ja paralleelsuse sisseehitatud tugi. Go populaarsus kasvab kiiresti, eriti pilvetehnoloogiate ja mikroteenuste vallas. [16]

Antud projekti jaoks on valitud Java, kuna see keel on hästisobiv serverirakenduse loomiseks oma skaleeritavuse, suure jõudluse ja ulatusliku tööriistakomplekti tõttu. Lisaks omab autor varasemat kogemust Java-ga, mis kiirendab arendusprotsessi ja vähendab vigade tekkimise riski.

### **3.3.3 Eesrakenduse tehnoloogiate valik**

JavaScript on saanud eelistatud keeleks eesrakenduse arenduses, kuna seda toetavad kõik kaasaegsed veebibrauserid. See võimaldab arendajatel rakendada interaktiivseid ja dünaamilisi funktsioone ilma täiendavate pistikprogrammide või tarkvarata. Lisaks on JavaScript arenenud kaugemale oma traditsioonilistest omadustest, tänu võimsatele raamistikutele nagu React, Angular ja Vue.js, mis pakuvad tugevaid tööriistu suuremahuliste ja tõhusate rakenduste loomiseks. Kõikide nende põhjuste tõttu oli JavaScript loogiline valik käesolevas lõputöös eesrakenduse arendamiseks, kusjuures Reacti kasutatakse ka selle efektiivse ja paindliku komponentide põhise arhitektuuri tõttu. [17]

TypeScript on oluline kaasaegses veebiarenduses, kuna see lisab JavaScriptile tugeva tüübikontrolli, mis aitab vigu varakult tuvastada ja parandada. Samuti toetab see omadusi nagu liidesed ja klassid, mis parandavad koodistruktuuri ja hooldatavust. Just nende põhjuste tõttu kasutatakse TypeScripti käesolevas lõputöös, et tagada puhtam ja tõhusam eesrakenduse arendus. [18]

Selliste tööriistade ja tehnoloogiate valik võimaldab saavutada kõrge jõudluse ja kasutusmugavuse arendustöös, mis on eriti oluline antud projekti jaoks.

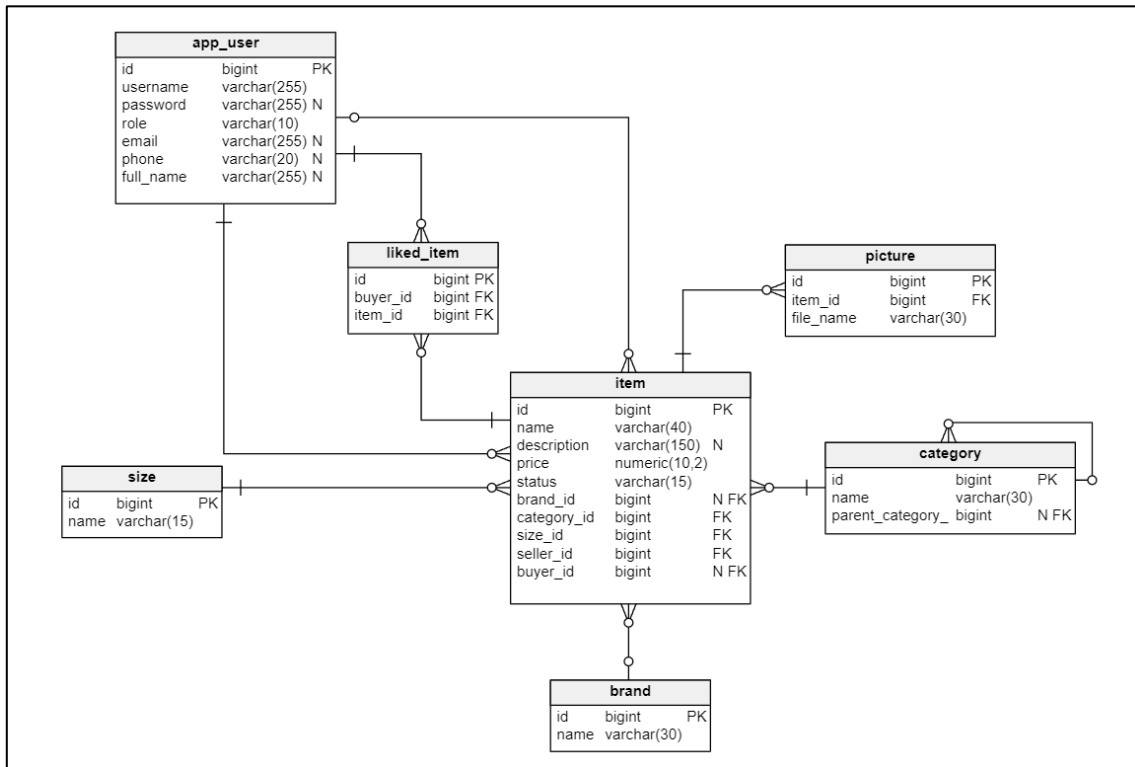
## **4 Veebirakenduse arendus**

Käesolevas peatükis käsitletakse veebirakenduse arendusprotsessi, hõlmates olulisi etappe, nagu andmebaasi projekteerimine ja loomine, serveripoolse osa arendus ning kasutajaliidese loomine. Erilist tähelepanu pööratakse kasutatud tehnoloogiatele, mis tagavad süsteemi kõikide komponentide funktsionaalsuse ja omavahelise koostöö.

### **4.1 Andmebaasi projekteerimine ja loomine**

Selle rakenduse aluseks on relatsiooniline andmebaas, mis mängib võtmerolli andmete salvestamises ja haldamises. Hästi struktureeritud andmebaas on veebirakenduse stabiilse toimimise alus, kuna see moodustab baasi, millele tuginevad kõik süsteemi komponendid (tagarakendus ja eesrakendus) oma funktsioonide täitmisel ja omavahelisel suhtlemisel.

Allpool joonisel 3 on esitatud iga andmebaasi tabeli funktsionaalne tähendus:



Joonis 3. Rakenduse andmemudel

- **app\_user**: See tabel haldab rakenduse kasutajate andmeid. Siin salvestatakse autentimiseks vajalik teave (kasutajanimi ja parool), kasutajate isikliku andmed (täisnimi, email, telefoninumber) ning nende roll süsteemis (näiteks administraator või tavakasutaja). Samuti on tabel aluseks kasutajate suhtlusele teiste süsteemi komponentidega
- **item**: Andmebaasi keskne tabel, mis täidab kaupade andmete salvestamise põhiülesannet. Selles tabelis on kõik kaupade omadused: nende nimetus, kirjeldus, hind, staatus ning seosed teiste tabelitega, nagu kategooriad, suurused, brandid ja pildid. Tabel jälgib ka seda, kes on toote müüja ja tulevikus ostja.
- **picture**: See tabel salvestab andmeid toodete piltide kohta. Siin on viited pildifailidele, et tagada toodete visuaalne esitus rakenduses. Iga pilt on seotud konkreetse tootega, mis võimaldab kasutajatele näidata vastavaid pilte iga eseme kohta.

- **category:** Vastutab kaupade organiseerimise eest kategooriatesse. Antud tabel võimaldab jagada tooteid kategooriatesse ning alamkategooriatesse. See lisab võimaluse mitmetasandiliseks sorteerimiseks, parandades seeläbi kasutajakogemust toodete otsingul.
- **size:** See tabel täidab kaupade saadaval olevate suuruste andmete salvestamise funktsiooni. See on standardne kategooriates, mille kasutades kasutajad saavad filtreerida tooteid suuruse järgi või valida vajaliku suuruse enne ostu sooritamist.
- **brand:** Tabel salvestab andmeid kaubamärkide kohta. Seda kasutatakse kaupade brändi määramiseks ja grupeerimiseks ning võimaldab kasutajatel filtreerida tooteid eelistatud brändide järgi.
- **liked\_item:** Tabel vastutab rakenduse "lemmikud" funktsiooni haldamise eest. See jälgib, milliseid tooteid kasutajad on lisanud lemmikute nimekirja, mis võimaldab pakkuda isikupärastatud kasutajakogemust ja kiiret juurdepääsu salvestatud või meeldivatele toodetele.

### **Andmebaasi kasutamine ja käivitamine**

Projektil kasutatakse PostgreSQL andmebaasi, mis on käivitatud läbi Dockeri. Fail `compose.yaml` laadib automaatselt PostgreSQL pildi ja käivitab konteineri eelnevalt määratud parameetritega (andmebaasi nimi, kasutaja, parool, port).

Tagarakendus suhtleb konteineriga, ühendudes andmebaasiga käivitamisel. See võimaldab automaatselt luua tabeleid või täita neid algandmetega, lihtsustades arenduse ja testimise protsessi.

## 4.2 Tagarakenduse arendus

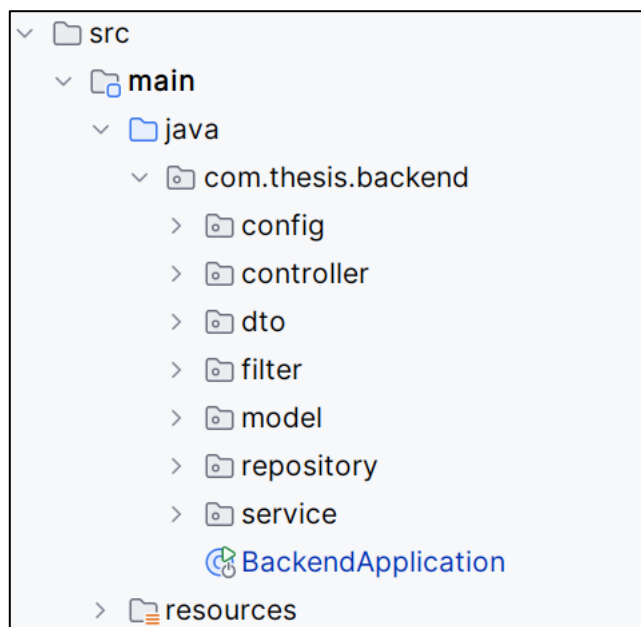
Tagarakenduse loomise eesmärk on arendada süsteemi serveripoolset osa, mis vastutab klientidelt saadud päringute töötlemise, andmete haldamise, nende salvestamise ja turvalisuse eest. See hõlmab serveri loogika väljatöötamist, andmebaaside loomist ja haldamist ning suhtlemist erinevate väliste teenuste ja API-dega. Tegelikult ühendab taagarakendus kasutajaliidese (eesrakenduse) ja andmebaasi, tagades nendevahelise info vahetamise ja äriloojika toimimise. [19]

Kuna tagarakenduse arendamiseks valiti Java programmeerimiskeel, otsustati kasutada populaarset raamistikku Spring Boot. Spring Boot muudab veebirakenduste ja teenuste loomise kiiremaks ja lihtsamaks, võimaldades arendajatel keskenduda rohkem rakenduse põhifunktsioonidele ja ideele, mitte infrastruktuuri seadistamisele.

Spring Boot'i üks suurimaid eeliseid on selle valmis konfiguratsioonid ja lai tööriistakomplekt, mis hõlmab näiteks sisseehitatud veebiservereid (nagu Tomcat), andmebaaside haldust ja turvafunktsioone. Lisaks pakub see skaleeritavust ja sobib ideaalselt nii väikestele kui ka suurtele projektidele, aidates arendajatel kiiresti prototüüpe luua ja neid vajadusel laiendada. [20]

### 4.2.1 Tagarakenduse ülesehitus

Joonisel 4 on kujutatud backend-rakenduse põhilist struktuuri, mis on loodud Spring Frameworki abil. See struktuur peegeldab klassikalist lähenemist serveripoolse rakenduse organiseerimisele, kus igal kaustal on kindel roll. Selline ülesehitus teeb projekti lihtsasti laiendatavaks, hooldatavaks ja mugavaks meeskonnatööks.



Joonis 4. Tagarakenduse projekti struktuur

Kaustade kirjeldus:

- **config:** Kaust konfiguratsiooniklasside jaoks, kus määratakse rakenduse seaded. Siia kuuluvad turvaseaded (SecurityConfig) ja CustomCorsConfiguration klass, mis tegeleb CORS-i (Cross-Origin Resource Sharing) seadistamisega, et määrata, millistest päritoludest päringud on rakendusele lubatud. See võimaldab rakendusel turvaliselt suhelda erinevate klientrakendustega.
- **controller:** Kontrollerid töötlevad sisenevaid HTTP-päringuid ja suunavad need vastavatele teenustele. Need toimivad kliendi ja serveripoolse loogika vahelise liidesena, rakendades REST API või muid suhtlustüüpe.
- **dto:** Selles kaustas asuvad klassid, mis on mõeldud andmete edastamiseks rakenduse kihtide vahel. DTO-d eraldavad sisemise andmemudeli kliendist, lihtsustades andmete teisendamist ja suurendades turvalisust.



- **filter:** Siin asuvad klassid, mis tegelevad päringute filtreerimisega, näiteks autentimise, autoriseerimise, logimise või päringutele päiste lisamisega. Selles projektis asub selles kaustas ainult JwtAuthenticationFilter, mis vastutab päringute autentimise eest, kontrollides ja valideerides JWT-tokeneid. See filter integreerub Spring Security-ga, tagades turvalise juurdepääsu rakenduse ressurssidele.
- **model:** Selles kaustas asuvad JPA-entiteedid, mis esindavad rakenduse andmeid. Need klassid on märgendatud ORM-i (Object-Relational Mapping) jaoks ja on otseselt seotud andmebaasitabelitega. Iga entiteet vastab andmebaasis olevale tabelile, võimaldades Springil hallata andmete salvestamist ja töötlemist automaatselt.
- **repository:** Kaust, kus asuvad liidesed ja klassid, mis töötavad andmebaasiga. Repositooriumid võimaldavad teostada CRUD-toiminguid (loomine, lugemine, uuendamine, kustutamine) Spring Data JPA abil, pakkudes lihtsat ja mugavat API-d.
- **service:** Selles kaustas asuvad klassid, mis vastutavad rakenduse äri loogika eest. Teenused ühendavad repositooriumite tööd ja lisavad andmetöötlust enne, kui need kontrollerrisse saadetakse.
- **resources:** Selles kaustas asuvad abimaterjalid ja olulised konfiguratsioonifailid. Siin paikneb application.yml, kus määratakse rakenduse peamised seaded, näiteks andmebaasi ühendus, serveri port ja muud konfiguratsiooniparameetrid. Lisaks sisaldab kaust faile, mis on seotud andmete valideerimisega ja algse täitmisega (schema ja data failid), mida kasutatakse andmebaasi skeemi loomiseks ja algandmete (seeding) lisamiseks. See tagab, et rakenduse andmebaas on õigesti üles ehitatud ja algväärtustatud.
- **BackendApplication:** Rakenduse peamine klass, mis sisaldab main() meetodit. See käivitab Spring Booti rakenduse, initsialiseerides rakenduse konteksti ja vajalikud komponendid.

Selline struktuur võimaldab tõhusalt jaotada vastutuse backend-arhitektuuri erinevate kihtide vahel, muutes rakenduse paindlikuks ja skaleeritavaks.

#### **4.2.2 Andmebaaside haldamine Spring JPA abil**

Java Persistence API (JPA) on spetsifikatsioon, mis pakub standardiseeritud viisi objekt-relatsioonilise kaardistuse (ORM) kasutamiseks Java-rakendustes. See võimaldab Java objekte andmebaasiridadeks teisendada ja vastupidi. JPA määratleb liidesed ja annotatsioonid, mis toetavad töötlust üksustega, vahemälu, tehinguid ja päringuid, kuid ei ole iseseisev teostus. Selleks on vajalikud sellised implementatsioonid nagu Hibernate.

Spring Data JPA on Spring Frameworki osa, mis lihtsustab JPA kasutamist, pakkudes kõrgetasemelist abstraktsiooni. See muudab andmetega töötamise efektiivsemaks, kuna arendajad saavad määrata meetodeid ainult nende nimede alusel samas kui Spring Data JPA genereerib vajalikud päringud automaatselt. See vähendab märgatavalt korduva koodi kirjutamise vajadust ja kiirendab arendust. Spring Bootiga integreerimine lihtsustab lisaks JPA konfigureerimist ja kasutuselevõttu tänu automaatsele seadistamisele.

Spring Data JPA toetab erinevaid päringumehhanisme, sealhulgas annotatsioone (@Query) käsitsi määratletud JPQL-päringute jaoks ja objektinäidiste kasutamist. See muudab Spring Data JPA väga paindlikuks ja tõhusaks tööriistaks andmetega töötamiseks, ühendades JPA paindlikkuse ja Spring Frameworki kasutajasõbralikkuse.

Käesolevas töös valiti selle tehnoloogia kasutamine andmebaasiga suhtlemiseks, kuna see lihtsustab ja kiirendab arendust, pakkudes samal ajal võimsaid ja paindlikke võimalusi andmetega töötamiseks. [21] [22]

#### **4.2.3 Rakenduse turvalisuse tagamine Spring Security abil**

Spring Security on võimas ja paindlik raamistik, mis on mõeldud Springi platvormil põhinevate rakenduste turvalisuse tagamiseks. See pakub tööriistu autentimiseks, autoriseerimiseks ja juurdepääsu haldamiseks ressurssidele.

Antud projektis kasutati autentimiseks JSON Web Tokeneid (JWT), mis on kompaktne viis andmete edastamiseks osapoolte vahel kodeeritud JSON-objekti kujul. JWT kasutamine Spring Securitys pakub järgmisi eeliseid:

- Lihtsustatud autentimine REST API-de jaoks: Erinevalt traditsioonilisest sessioonipõhisest lähenemisest ei nõua JWT serveripoolset oleku salvestamist.

- Turvalisus: Igal JWT-l on allkiri, mis on loodud salajase võtme abil. See tagab, et tokenit ei saa võltsida seni, kuni salajane võti jääb konfidentsiaalseks.
- Laiendatavus: JWT-d võivad sisaldada kohandatud välju (nt kasutaja rollid või identifikaatorid), mis võimaldab süsteemil teha autoriseerimisotsuseid ilma täiendavate andmebaasi päringuteta.

Spring Security kaitseb ka levinud rünnakute eest ja tänu seadistuste paindlikkusele on võimalik kasutada nii standardseid kaitsemehhanisme kui ka rakendada oma reegleid juurdepääsu ja tokenite valideerimise jaoks. [23]

#### **4.2.4 AWS S3 piltide üles- ja allalaadimiseks**

Veebirakenduses otsustati kasutada AWS S3 teenust piltide salvestamiseks. Piltide üles- ja allalaadimise protsess toimub, kui front-end saadab päringu back-endile, mis genereerib eelmääratud allkirjastatud lingid (pre-signed URL-id). Selle lahenduse rakendamiseks oli kasutatud AWS SDK, mis võimaldab kasutada spetsiaalset klassi vajalike linkide genereerimiseks. Eelnimetatud linkide abil saab front-end otse üles- ja allalaadida pilte S3-st, tagades samas turvalisuse ning vähendades serveri koormust.

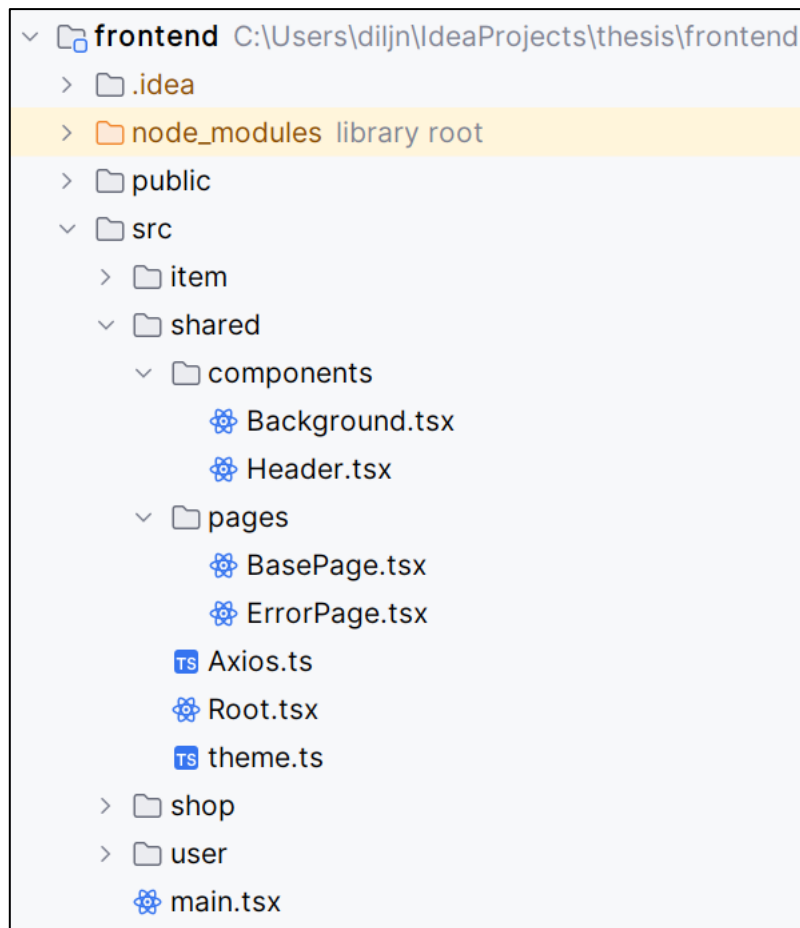
### **4.3 Kasutajaliidese loomine**

Selles peatükis käsitletakse lõputöö eesrakenduse osa. Esmalt tutvustatakse ja selgitatakse eesrakenduse struktuuri, sealhulgas selle põhielemente, nende korraldust ja otstarvet. Seejärel tutvustatakse kättesaadavat funktsionaalsust, mis kirjeldab kasutajaliidese peamisi võimalusi ja sellega seotud interaktsioone.

#### **4.3.1 Eesrakenduse struktuuri ülevaade**

Projekt on loodud, kasutades TypeScript koos React teegiga. Koodi struktuuri lihtsustamiseks ja selguse tagamiseks otsustas autor kasutada modulaarset ülesehitust. Iga rakenduse funktsionaalne osa on kujundatud eraldi moodulina, mis lihtsustab koodi hooldamist, laiendamist ja taaskasutamist.

Joonisel 5 on esitatud eesrakenduse struktuur:



Joonis 5. Eesrakenduse projekti struktuur

Joonisel on näidatud moodul *shared*, mis sisaldab kogu rakenduses kasutatavaid ühiseid ressursse ja loogikat. Selle sees on tüüpiline struktuur, mis hõlmab järgmisi elemente:

- *components* – kaust, mis sisaldab komponente, mida kasutatakse rakenduse erinevates osades. Näiteks pildil moodulis *shared/components* asuvad üldised komponendid nagu *Header.tsx* ja *Background.tsx*. Need komponendid võimaldavad taas kasutada valmis visuaalseid elemente ja lihtsustavad kasutajaliidese hooldust.
- *pages* – kaust, kuhu salvestatakse lehed, mis määravad kasutajaliidese põhivaated. *BasePage.tsx* on baasmall lehele, samas kui *ErrorPage.tsx* kuvatakse vigade korral.
- *api.ts* – fail, kuhu salvestatakse funktsioonid serveri poolega suhtlemiseks. Tavaliselt sisaldab see HTTP-päringute loogikat, näiteks andmete toomiseks või saatmiseks.

- *dto* – fail, kus säilitatakse DTO objektid, mida kasutatakse klient-server suhtluses andmete struktuuri määratlemiseks. See aitab standardiseerida andmete formaati ja parandab nende töötlemist.

Teised moodulid, nagu *item*, *shop* ja *user*, on üles ehitatud sarnaselt. Igal neist võivad olla oma lehed, komponendid, API-loogika ja DTO-failid, mis on seotud konkreetse mooduli funktsionaalsusega.

Moodulis *shared* mängivad olulist rolli järgmised failid:

- *Axios.ts* – HTTP-kliendi Axios konfiguratsioonifail. See klient võimaldab serverile päringute saatmist, ja konfiguratsioon lisab automaatselt JWT-tokeni igale päringule, tagades sellega kasutaja autoriseerimise.
- *Root.tsx* – mooduli põhikomponent, mis ühendab ühised kasutajaliidese elemendid, näiteks päised ja taustaelemendid.

Rakenduse ühtse ja kaasaegse kujunduse loomiseks kasutatakse Material-UI teeki. Moodulis *shared* asuvas failis *theme.ts* hoitakse teema konfiguratsiooni, sealhulgas värvipaletti ja kasutajaliidese elementide stiile. See lahendus võimaldab paindlikult hallata kogu rakenduse kujundust ja hoida ühtset disaini.

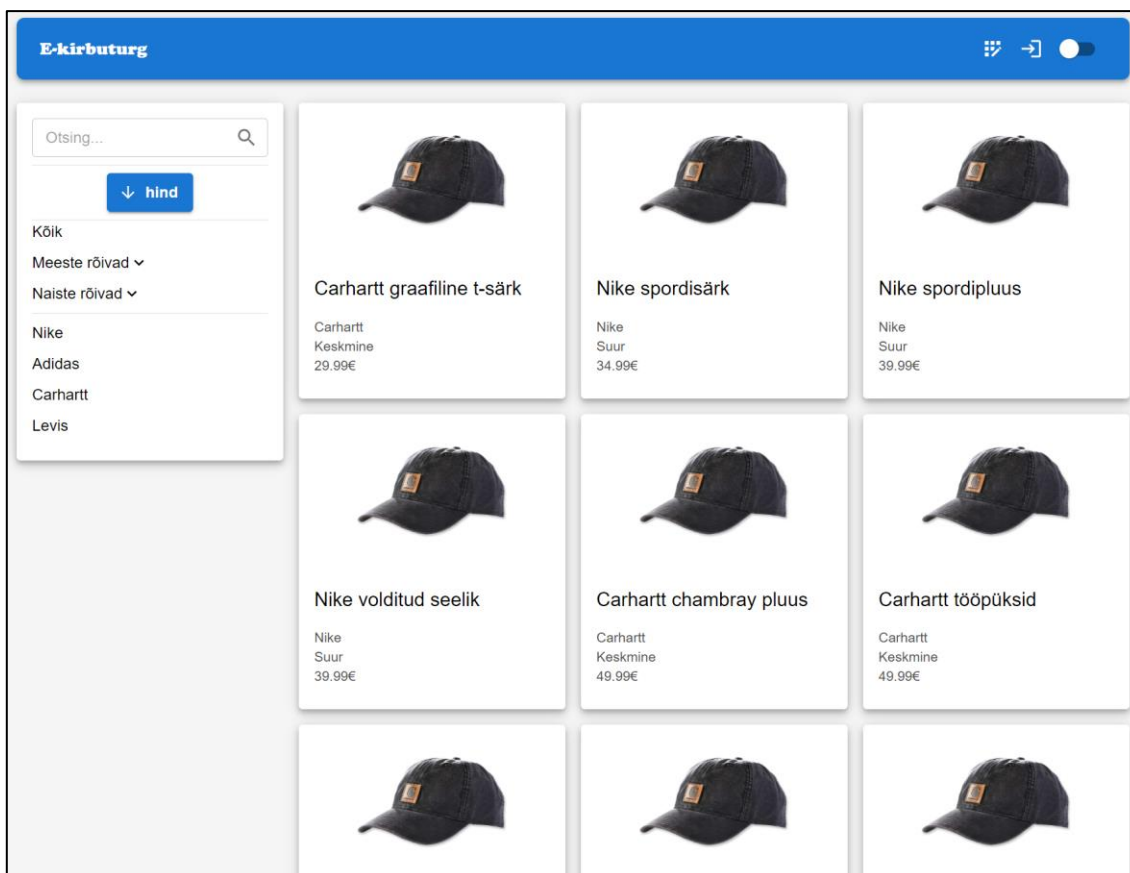
Fail *main.tsx* vastutab rakenduse algatamise ja ruuteri seadistamise eest. Ruuter määrab, milliseid lehti kasutajale näidata, sõltuvalt sisestatud URL-ist. See võimaldab korraldada rakenduse navigeerimist ja ühendada vastavad moodulid ning nende lehed.

### 4.3.2 Loodud kasutajaliidese ja selle funktsionaalsuse ülevaade

Käesolevas peatükis on esitatud loodud veebirakenduse funktsionaalsed võimalused ja visuaalsed elemendid. Illustratsioonidel olevad kaupade pildid on ühesugused, kuna autor kasutas piiratud juurdepääsu AWS S3 ressurssidele (tasuta pakett), et minimeerida kulusid.

Veebirakenduse avalehel (kodulehel) (Joonis 6) saab kasutaja näha kõiki sellel müügiplatvormil saadaolevaid tooteid. Otsingu lihtsustamiseks on lehe vasakul küljel otsingupaneel, kus asuvad otsinguriba ning toodete kategooriad ja brändid. Nende tööriistade abil saab kasutaja filtreerida tooteid kategooriate, brändide, suuruste või nimede järgi.

Lehe ülaosas asub sinine navigeerimisriba, mis võimaldab kasutajal registreeruda või sisse logida. Täiendava funktsioonina on paremas ülanurgas lüliti, mis võimaldab vahetada heleda ja tumeda teema vahel, parandades kasutajaliidese isikupärastamise võimalusi.



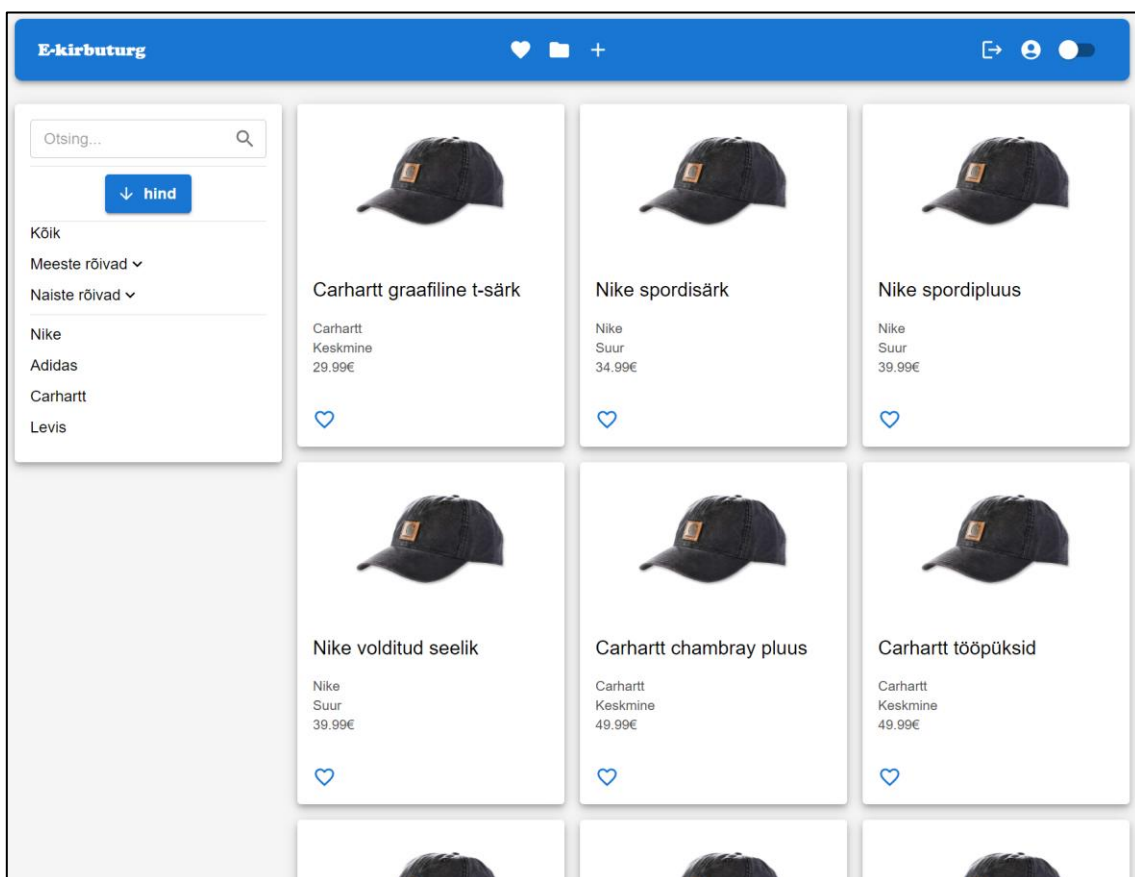
Joonis 6. Rakenduse avaleht

Pärast autoriseerimist või registreerimist avaneb kasutajale lisafunktsionaalsus (Joonis 7). Avalehel toodet esindaval kaardil muutub võimalikuks lisada toode lemmikute nimekirja.

Lisaks ilmuvad navigeerimispaneeli keskele kolm ikooni. Esimene ikoon võimaldab vaadata salvestatud lemmiktooteid, teine ikoon võimaldab vaadata kasutaja enda lisatud tooteid ning kolmas annab võimaluse luua uus kuulutus.

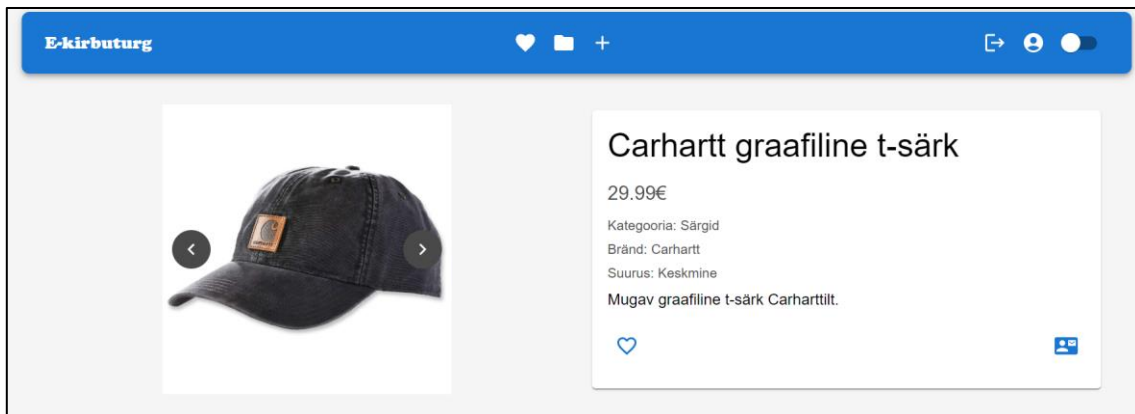
Lemmiktoodete ja kasutaja enda toodete lehed on kujunduselt sarnased avalehega, kuid neil puudub külgpaneel. Lisaks on nendel lehtedel kuvatavad tooted automaatselt filtreeritud vastavalt kasutaja lemmikutele või tema lisatud toodetele, mis tagab selguse ja lihtsuse toodete haldamisel.

Kasutusmugavuse ja selguse parandamiseks kuvatakse kõigi navigeerimispaneelil asuvate ikoonide kohal hiirega liikudes vastavad selgitavad tööriistad.



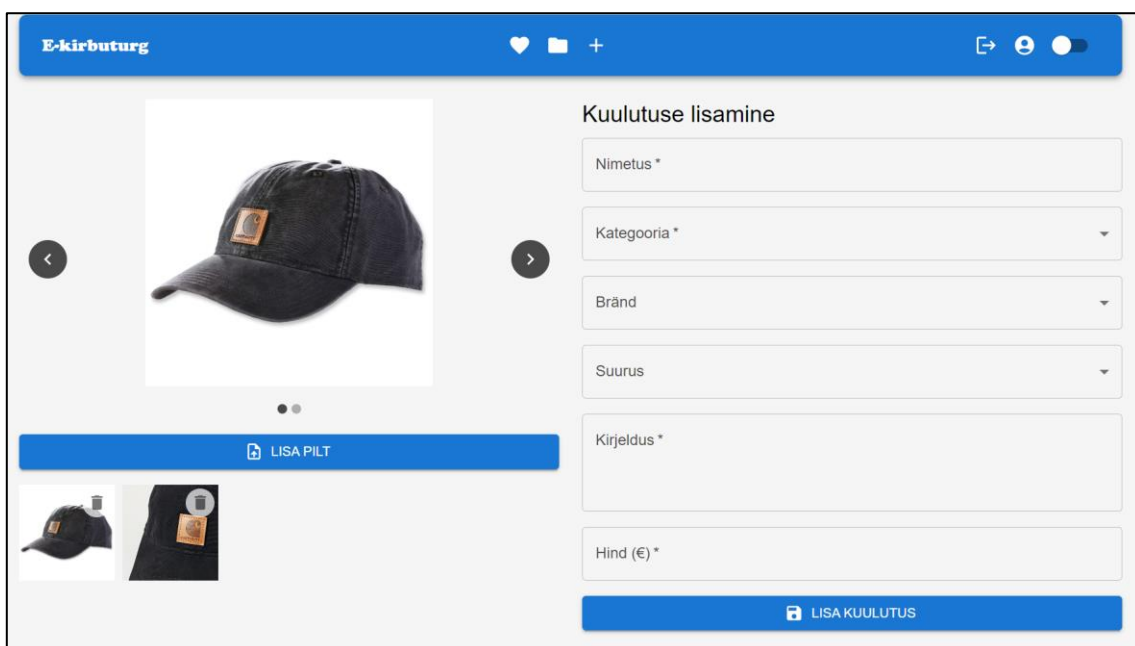
Joonis 7. Autoriseeritud kasutaja avaleht

Klõpsates tootele, avaneb lehekülg, kus kuvatakse kogu salvestatud teave toote kohta. Sellel lehel saavad kasutajad vaadata müüja lisatud tootepilte. Varem autoriseeritud kasutajatele on lisaks saadaval võimalus salvestada toode oma lemmikute nimekirja ning vaadata müüja kontaktandmeid, et võtta müüjaga ostu osas ühendust. (Joonis 8)



Joonis 8. Toote detailvaade

Toodete lisamise ja muutmise vaade sarnaneb detailvaatega. Sellel lehel saab müüja sisestada või muuta vajaliku teabe toote kohta. (Joonis 9)



Joonis 9. Toote lisamise ja muutumise vaade



## 4.4 Prototüübi testimine

Rakenduse prototüübi hindamiseks viidi läbi testimine, kus osales 10 tudengit, kelle ülesandeks oli analüüsida veebirakenduse kasutajakogemust, funktsionaalsust ning teha ettepanekuid täiustuste osas. Testijatel paluti hinnata põhifunktsioonide toimimist, sealhulgas toodete otsingut, lemmikute lisamist, uute kuulutuste loomist ning olemasolevate muutmist. Testijad pakkusid mitmeid väärtuslikke täiendusettepanekuid, mis võiksid rakenduse kasutajakogemust veelgi parendada. Need ideed ja soovitused puudutasid erinevaid aspekte, andes suunda, kuidas arendustegevust tulevikus suunata ja rakendust täiendavalt täiustada.

Testimise käigus ilmnisid mitmed tehnilised puudused, mis vajavad parandamist. Peamiseks probleemiks osutus lemmikute nupu funktsionaalsus, mis ei töötanud alati ootuspäraselt. Samuti esines mõningaid vigu kuulutuste redigeerimisel, mis põhjustasid ootamatuid katkestusi. Selliste tehniliste tõrgete kõrvaldamine on oluline, et tagada rakenduse sujuv ja usaldusväärne toimimine.

Vaatamata tuvastatud puudustest hindasid testijad rakendust üldiselt positiivselt. Nad tõstsid esile selle kaasaegse ja kasutajasõbraliku disaini. Rakendust peeti kergesti navigeeritavaks ning selle funktsionaalsust kiideti. Testijate arvates on rakendusel olemas potentsiaal ning täienduste elluviimine tõstaks selle kvaliteeti veelgi. Edasised arendusetapid keskenduvad nende parandusettepanekute rakendamisele, et pakkuda tulevastele kasutajatele veelgi paremat kogemust ja saavutada rakenduse täielik potentsiaal.

## 4.5 Edasise arendamise võimalused

Prototüübi testimise käigus ilmnisid mitmed tehnilised probleemid, kuid samas toodi esile ka mitmeid olulisi täiendustettepanekuid, mis võiksid rakenduse kasutajakogemust veelgi parandada.

Esimene soovitus oli Google'i konto lihtsustatud autentimise lisamine, et muuta registreerumis- ja sisselogimisprotsess kiiremaks ja mugavamaks, mis saab pakkuda sujuvamat ja kiiremat kasutajakogemust.

Samuti töid testijad välja vajaduse laiendada rakenduses saadaolevate brändide ja suuruse valikut, et pakkuda kasutajatele mitmekesisemat ostukogemust. Nende laiendamine aitaks katab rohkem kasutajate eelistusi ja suurendaks rakenduse atraktiivsust, pakkudes laiemat valikut erinevaid tooteid.

Autori ettepanekud rakenduse edasisteks täiustusteks tulevikus on järgmised:

- Maksesüsteemi integreerimine, et muuta ostuprotsess kiiremaks ja turvalisemaks. Kasutajad saaksid maksta otse rakenduse kaudu, mis suurendaks ostumugavust.
- Sõnumivahetuse võimalus rakenduses, mis parandaks kasutajatevahelist suhtlust. See funktsioon võimaldaks müüjatel ja ostjatel suhelda otse rakenduse kaudu, pakkudes paremat kasutajatuge ja kiirendades müügiprotsessi.
- Täiendatud sorteerimissüsteem, lisades rohkem suuruseid, kategooriaid ja nendevahelisi seoseid. Näiteks valides jalatsite kategooria, kuvatakse kasutajale ainult jalatsite suurused. See muudaks sorteerimist veelgi intuitiivsemaks ja täpsemaks.

Need täiendustettepanekud võimalikult aitavad suurendada rakenduse funktsionaalset pakkumist, pakkuda kasutajatele paremat müügi- ja ostuteenust ning tagada sujuvam ja meeldivam kasutajakogemus. Edasised arendusetapid on suunatud nende ideede elluviimisele, et saavutada täiendav areng ja kvaliteedi tõus.

## 5 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli arendada veebipõhise müügiplatvormi prototüüp, mis võimaldab kasutajatel müüa ja osta taaskasutatud rõivaid. Projekti keskne eesmärk oli luua lahendus, mis ühendaks ostjad ja müüjad ühtseks turuks, pakkudes lihtsat ja usaldusväärset viisi kaupade ostmiseks ja müümiseks, kusjuures platvormi kasutamise eesmärk on lihtsustada ja ühendada kogu ostu-müügi protsess.

Käesoleva töö raames analüüsiti kaasaegseid maailma probleeme ja tendentse, mis on seotud taaskasutatud rõivastega ning nende kasvavate võimalustega sekundaarturul. Samuti uuriti Eestis juba olemasolevaid veebipõhiseid lahendusi rõivaste müümiseks ja ostmiseks, hinnates nende tugevusi ja nõrkusi. Selle esmase analüüsi ning autori kogemuste ja ressursside põhjal määratleti töö maht ja skoop.

Töö tulemusena valmis veebipõhine rakendus, mis vastas kõigile arenduse käigus püstitatud nõudmistele ning sisaldas kogu vajalikku minimaalset funktsionaalsust. Rakenduse loomisel lähtuti nõuete analüüsist, rakenduse arhitektuurist ja selle kihtidest, samuti kasutati tööriistu, nagu programmeerimiskeeled ja raamistikud. Valitud arhitektuur ja tehnoloogiad tagasid rakenduse töökindluse toimimise, võimaldades luua funktsionaalse lahenduse, mis toetab projekti eesmärki pakkuda lihtsat ja tõhusat keskkonda taaskasutatud rõivaste ostmiseks ja müümiseks.

Rakenduse testimisel leiti, et see täidab oma põhifunktsioonid tõhusalt, kuid mõned täiustused on veel vajalikud, et rakendus oleks valmis täisfunktsionaalseks avalikuks kasutamiseks. Testimise käigus leiti, et rakendusel on olemas potentsiaal ning annab head lähtepunktid edasiseks arendamiseks, et täiendada funktsioone nagu integreeritud maksusüsteem, sisselogimine Google kontoga ning täiendatud ja keerukam sorteerimise ja filtreerimise funktsionaalsus.

Kokkuvõttes lahendab loodud rakendus püstitatud probleemi, pakkudes lihtsat ja tõhusat lahendust kasutatud kaupade müügiks ja ostmiseks. Prototüüp täidab kõik vajalikud ülesanded testimise tasemel, kuid edasised arendused ja parendused aitaksid rakendusel saavutada selle täieliku potentsiaali ja valmisoleku avalikuks kasutamiseks. Kogu töö maht, sealhulgas arendus, analüüs ja prototüübi loomine, viidi läbi autori poolt.

## Kasutatud kirjandus

- [1] S. Mutreja, „The cost-of-living crisis has accelerated the move to second hand,“ [Võrgumaterjal]. Available: <https://www.cardlytics.com/blog/the-cost-of-living-crisis-has-accelerated-the-move-to-second-hand>. [Kasutatud 02 11 2024].
- [2] „2022 Fashion Resale Market and Trend Report | thredUP,“ ThredUp, 2022. [Võrgumaterjal]. Available: <https://www.thredup.com/resale/2022/#size-and-impact>. [Kasutatud 02 12 2024].
- [3] C. Swanson, „Social Media Shifts Trends in Thrift and Resale Culture,“ Buzzsaw , 20 05 2020. [Võrgumaterjal]. Available: <https://buzzsawmag.org/2020/05/20/social-media-shifts-trends-in-thrift-and-resale-culture/>. [Kasutatud 02 11 2024].
- [4] „Sustainability starts with students: Gen Z’s second-hand shopping trend,“ The Social Institute, 13 04 2023. [Võrgumaterjal]. Available: <https://thesocialinstitute.com/blog/sustainability-starts-with-students-gen-zs-second-hand-shopping-trend/>. [Kasutatud 02 12 2024].
- [5] Vincent, „The Sustainability of Thrifting: How Secondhand Shopping Reduces Your Environmental Impact,“ Thrift City, 2023. [Võrgumaterjal]. Available: <https://thriftcityllc.com/sustainability-of-thrifting-reduces-environmental-impact/>. [Kasutatud 02 11 2024].
- [6] „Instagram Shopping,“ 10 06 2024. [Võrgumaterjal]. Available: <https://sendpulse.com/support/glossary/instagram-shopping>. [Kasutatud 02 11 2024].
- [7] D. Dragilev, „Top 10 Disadvantages of an Instagram Business Account in 2023,“ 17 01 2023. [Võrgumaterjal]. Available: <https://smallbiz.tools/disadvantages-of-an-instagram-business-account/>. [Kasutatud 05 11 2024].
- [8] „Kuidas osta kasutatud asju turvaliselt?,“ 17 11 2021. [Võrgumaterjal]. Available: <https://medium.com/@Ostuportaamid/kuidas-turvaliselt-osta-kasutatud-kaupa-4add738b761b>. [Kasutatud 12 11 2024].
- [9] H. Nguyen, „The In-Depth Facebook Marketplace Reviews,“ 06 2024. [Võrgumaterjal]. Available: <https://litcommerce.com/blog/facebook-marketplace-reviews/>. [Kasutatud 06 11 2024].
- [10] „An Overview of the Estonian eCommerce Sector,“ International Trade Council, 05 06 2022. [Võrgumaterjal]. Available: <https://tradecouncil.org/backup/an-overview-of-the-estonian-ecommerce-sector/>. [Kasutatud 02 12 2024].
- [11] P. Kalda, „Vaata, kuidas vältida osta.ee-s pettasaamist,“ 08 09 2015. [Võrgumaterjal]. Available: <https://arileht.delfi.ee/artikkel/72395451/vaata-kuidas-valtida-ostaee-s-pettasaamist>. [Kasutatud 02 12 2024].
- [12] P. Gupta, „The pros and cons of a layered architecture pattern,“ 12 12 2023. [Võrgumaterjal]. Available: <https://www.techtarget.com/searchapparchitecture/tip/The-pros-and-cons-of-a-layered-architecture-pattern>. [Kasutatud 17 11 2024].
- [13] TestGorilla, [Võrgumaterjal]. Available: <https://www.testgorilla.com/blog/sql-vs-nosql/>. [Kasutatud 02 12 2024].

- [14] M. Smallcombe, „SQL vs NoSQL: 5 Critical Differences,“ 15 02 2024. [Võrgumaterjal]. Available: <https://www.integrate.io/blog/the-sql-vs-nosql-difference/#:~:text=SQL%20databases%20are%20table%2Dbased,data%20like%20documents%20or%20JSON>. [Kasutatud 21 11 2024].
- [15] „DB-Engines Ranking - Trend Popularity,“ DB-engines, 12 2024. [Võrgumaterjal]. Available: [https://db-engines.com/en/ranking\\_trend](https://db-engines.com/en/ranking_trend). [Kasutatud 02 12 2024].
- [16] T. Tran, „Top Server-side Scripting Languages & Frameworks,“ 27 05 2024. [Võrgumaterjal]. Available: <https://www.orientsoftware.com/blog/server-side-scripting-languages/>. [Kasutatud 02 12 2024].
- [17] „JavaScript in Front-end Development: The Ultimate Guide to Industrial Benefits,“ Hire Independent Developers, [Võrgumaterjal]. Available: <https://www.hireindependentdevelopers.com/blog/javascript-in-front-end-development/>. [Kasutatud 02 12 2024].
- [18] SOVANNARO, „What is TypeScript and Why Should You Use It?,“ 01 12 2021. [Võrgumaterjal]. Available: <https://dev.to/sovannaro/what-is-typescript-and-why-should-you-use-it-1558>. [Kasutatud 02 12 2024].
- [19] „Back-End Web Architecture,“ Codecademy, [Võrgumaterjal]. Available: <https://www.codecademy.com/article/back-end-architecture#:~:text=The%20back%2Dend%20is%20the,the%20data%20for%20the%20application..> [Kasutatud 28 11 2024].
- [20] „Why Spring?,“ Spring, [Võrgumaterjal]. Available: <https://spring.io/why-spring>. [Kasutatud 28 11 2024].
- [21] L. Agarwal, „Magic Behind Spring Data JPA: Simplifying Database Access in Java,“ 20 01 2024. [Võrgumaterjal]. Available: <https://medium.com/@lakshyachampion/magic-behind-spring-data-jpa-simplifying-database-access-in-java-89d87224d84a>. [Kasutatud 30 11 2024].
- [22] „Spring Boot JPA,“ Javatpoint, [Võrgumaterjal]. Available: <https://www.javatpoint.com/spring-boot-jpa>. [Kasutatud 30 11 2024].
- [23] M. Behler, „Spring Security: Authentication and Authorization In-Depth,“ 30 05 2022. [Võrgumaterjal]. Available: <https://www.marcobehler.com/guides/spring-security#oauth2>. [Kasutatud 30 11 2024].

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Daniil Iljin

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Müügiplatvorm veebirakendusena rõivaste sekundaarturu toetamiseks“, mille juhendaja on Lembit Viilup
  - 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. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

03.12.2025

---

<sup>1</sup> 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 loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## **Lisa 2 – rakenduse repositoorium**

Repositooriumi link: <https://github.com/DaniilIjin/thesis>