Tallinna Tööstushariduskeskus  
Noorem Tarkvaraarendaja erialal

**TARKVARAARENDUS “SmartStock” LOOMINE**   
**PRAKTILINE TÖÖ**

Õpilased: Maksim Tšepelevitš  
Juhendaja: kutseõpetaja Marina Oleinik

Tallinn 2024

Sisukord

[1 Sissejuhatus 2](#_Toc184164341)

[1.1 Sihtgrupp 2](#_Toc184164342)

[2 Teoreetiline osa 4](#_Toc184164343)

[2.1 Tehnilised nõuded 4](#_Toc184164344)

[**Visual Studio 2022** 4](#_Toc184164345)

[2.2 Funktsionaalsus ja kasutajaliides 5](#_Toc184164346)

[2.3 Kasutajaliidese nõuded 6](#_Toc184164347)

[2.4 Turvanõuded 6](#_Toc184164348)

[2.5 Tulemuslikkuse nõuded 6](#_Toc184164349)

[2.6 Muud mittefunktsionaalsed nõuded 7](#_Toc184164350)

[2.7 Nõuete täitmise kirjeldus 7](#_Toc184164351)

[2.8 Projekti ülevaade 8](#_Toc184164352)

[3 Praktiline osa 9](#_Toc184164353)

[3.1 Mudeli loomine 10](#_Toc184164354)

[3.2 Andmebaasi loomine 11](#_Toc184164355)

[3.3 Backend-i loomine 12](#_Toc184164356)

[3.4 Frontend-i loomine 13](#_Toc184164357)

[3.5 Testimine ja silumine 14](#_Toc184164358)

[**3.5.1** Testimine Lokaalses Keskkonnas (localhost) 14](#_Toc184164359)

[**3.5.2** Valmisoleku Testimine 14](#_Toc184164360)

[3.6 Testimis kokkuvõte 15](#_Toc184164361)

[4 Kasutusjuhend 16](#_Toc184164362)

[4.1 Pealeht 16](#_Toc184164363)

[4.2 Kastutaja poolt 19](#_Toc184164364)

[4.3 Administraatori poolt 22](#_Toc184164365)

[5 Kokkuvõte 26](#_Toc184164366)

[6 Allikad 27](#_Toc184164367)

# Sissejuhatus

**Projekt: "SmartStock" –** platvorm, mis võimaldab hallata inventari ja tooteid**.**

**Teema valiku põhjendus:**

Tänapäeva maailmas seisavad ettevõtted ja organisatsioonid silmitsi vajadusega tõhusalt  
hallata tooteid ja inventari. Projekt "SmartStock" lahendab selle ülesande, pakkudes kasutajatele platvormi toodete haldamiseks, tellimuste ja tarnete jälgimiseks. Loome süsteemi, mis lihtsustab seda protsessi, muutes selle kõigile kasutajatele kergesti kergesti arusaadavaks ja juurdepääsetavaks.

**Projekti eesmärk:** "SmartStock" pakub lihtsat viisi inventari ja tootevarude haldamiseks, pakkudes järgmisi võimalusi:

* Toodete ja nende varude haldamine.
* Tellimuste loomine ja jälgimine.
* Tarnete jälgimine.
* Lihtne suhtlus tellijate ja tarnijate vahel.

Aktuaalsus, tähtsus ja vajadus: Digitaalsete tehnoloogiate arenguga muutub selgeks vajadus automatiseerida protsessid, mis on seotud toodete ja inventariga. "SmartStock" pakub kasutajale mugavat platvormi nende ülesannete lihtsustamiseks, efektiivsuse suurendamiseks ja varude arvestuse ja töötlemise protsesside kiirendamiseks.

## Sihtgrupp

Projekt **"SmartStock"** on suunatud järgmistele kasutajate gruppidele:

* Tellijad, kes soovivad oma tooteid jälgida, teha tellimusi ja neid hallata.
* Tarnijad, kes soovivad tõhusalt hallata tarnet ja jälgida oma tooteid platvormil.

**Riistvara ja tarkvara:** Visual Studio 2022, PHPStorm, React.js, ASP.NET MVC, C#, Entity Framework, MySQL, Owin, Autofac, Autofac.Mvc 5

# Teoreetiline osa

## Tehnilised nõuded

**Visual Studio 2022**

Visual Studio on valitud peamiseks arenduskeskkonnaks backend arenduseks. See pakub tööriistu kiireks ja efektiivseks C# arendamiseks koos ASP.NET MVC integreerimisega.

**PHPStorm**

PHPStorm on valitud frontend arenduseks, kuna see pakub head toega React.js ja muudab arenduse kiireks ja intuitiivseks.

**React.js**

Valitud React.js-i põhjalik komponent-põhine struktuur, et luua dünaamiline ja kiire kasutajaliides, millel on tugev toetus kasutaja tegevustele.

**ASP.NET MVC**

Kasutatud framework backend arendamiseks, et hallata äriloogikat ja hallata andmeid ning aidata eraldada äriloogikat, vaateid ja kontrollereid.

**C#**

C# on valitud keeleks oma kõrge jõudluse, paindlikkuse ja tugeva objektorienteeritud programmeerimise toe tõttu.

**Entity Framework**

ORM tööriist, mis lihtsustab andmebaasi haldamist, võimaldades andmebaasiga töötada objekti-põhise lähenemisega.

**MySQL**

Kasutatakse andmete salvestamiseks ja haldamiseks usaldusväärse andmebaasisüsteemina.

**Owin**

Tagab veebirakenduse ja serveri vaheline abstraktsiooni, pakkudes suuremat paindlikkust rakenduste ja serverite haldamiseks.

**Autofac**

Kasutatakse sõltuvuste haldamiseks ja sõltuvuste süstimiseks backend rakendustes, et lihtsustada objekti elutsükli haldamist ja vähendada koodikeerukust.

## Funktsionaalsus ja kasutajaliides

**Administraatori funktsioonid**

* + Kasutajate ja toodete haldamine (lisamine, muutmine, kustutamine).
  + Tarnijate kinnitamine.

**Tellija funktsioonid**

* + Toodete ja varude haldamine (lisamine, vaatamine, tellimine).
  + Tellimuste loomine ja jälgimine.
  + Tellimuste andmete eksportimine.

**Tarnija funktsioonid**

* + Tarnete haldamine.
  + Tarnete andmete eksportimine.
  + Kaupade vastuvõtmise ja saatmise kinnitamine.

## Kasutajaliidese nõuded

* **Intuitiivsus:** Kõik kasutajaliidesed peavad olema lihtsad ja arusaadavad igale kasutajale: administraator, tellija või tarnija.
* **Kasutusmugavus:** Lihtne navigeerida ja teha soovitud toimingud.
* **Juhtimine ja disain:** Lihtne ja intuitiivne navigeerimisstruktuur.
* **Ligipääsetavus:** Rakendus peab järgima WCAG (Web Content Accessibility Guidelines) ligipääsetavuse standardeid.
* **Kohanduvus:** Rakendus peab olema kohandatud erinevatele seadmetele (mobiiltelefonid, tahvelarvutid, lauaarvutid).

## Turvanõuded

* **Autentimine ja autoriseerimine:** Kasutajad peavad olema autentitud, et süsteemi kasutada, kasutades sisselogimis- ja parooli meetodeid.
* **Andmete turvalisus:** Kõik tundlikud andmed peavad olema krüpteeritud nii puhkeolekus kui ka edastamisel.
* **Rollipõhine ligipääs:** Erinevad ligipääsuõigused (administraator, tellija, tarnija), et määrata, milliseid funktsioone ja andmeid kasutajad saavad vaadata või muuta.

## Tulemuslikkuse nõuded

* **Laadimiskiirus:** Lehekülgede ja materjalide laadimiskiirus peaks olema alla 2 sekundi.
* **Mastaapsus:** Süsteem peaks suutma toetada suurt hulka kasutajaid ja tooteid korraga, ilma et see mõjutaks jõudlust.
* **Usaldusväärsus:** Süsteem peaks olema vähemalt 99,9% ajast saadaval.

## Muud mittefunktsionaalsed nõuded

* **Logimine ja jälgimine:** Kõik süsteemi tegevused (nt kasutajate tegevused, vead) peavad olema logitud ja administraatoril ligipääsetavad.
* **Varundamine ja taastamine:** Andmed (nt tooted, kasutajad, tellimused) peavad olema regulaarselt varundatud ja taastamise võimalus peab olema tagatud.
* **Väliste teenuste integratsioon:** Väliste teenuste integreerimise võimalus, nagu maksesüsteemid ja SMS-teenused.

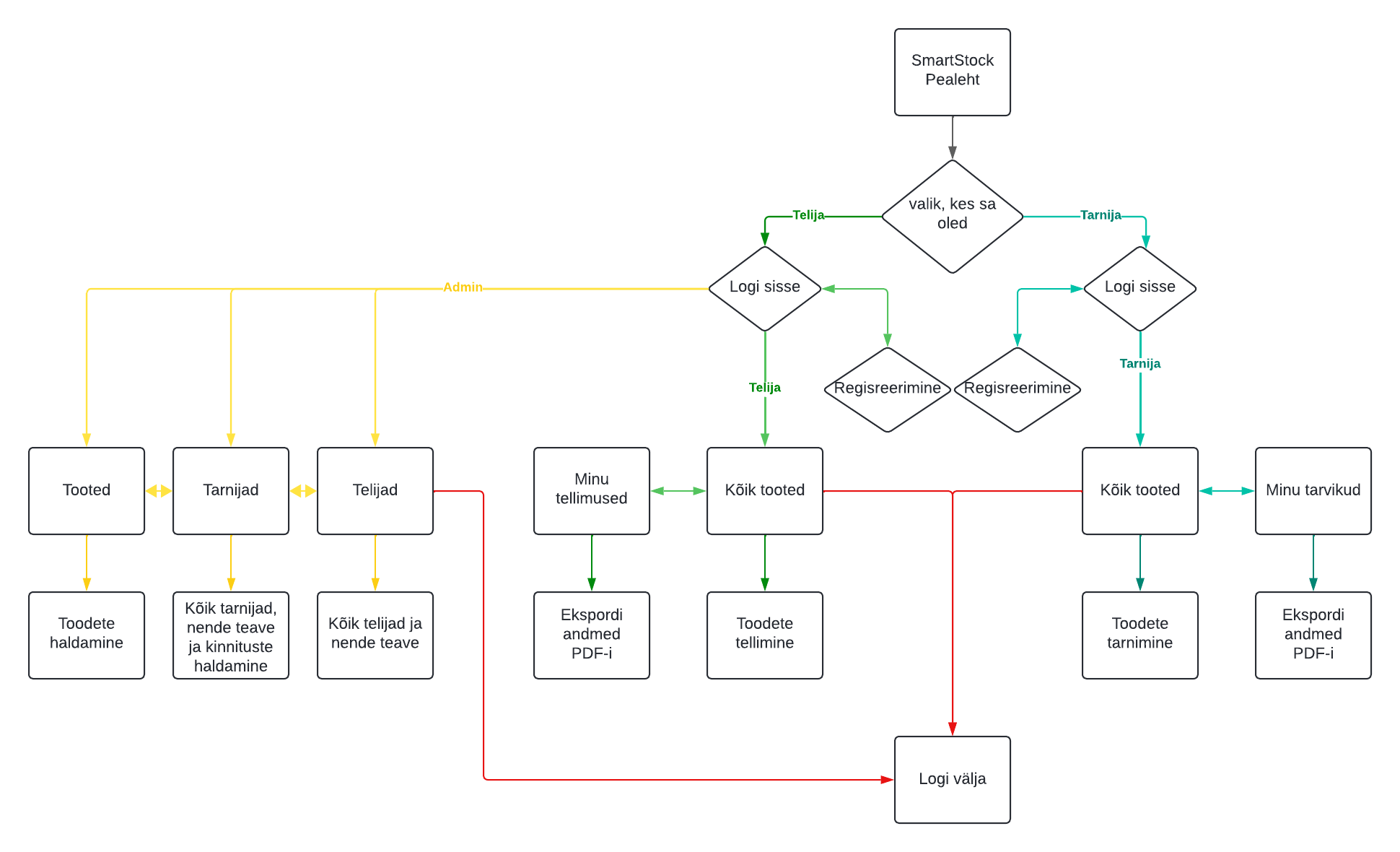
## Nõuete täitmise kirjeldus

**Funktsionaalsete nõuete täitmine:** Projekt täidab kõik funktsionaalsed nõuded, pakkudes selgeid kasutajaliideseid kolmele kasutajarollile: administraator, tellija ja tarnija. Kasutajad saavad hallata tooteid, luua tellimusi ja jälgida nende olekuid. Administraatoril on juurdepääs süsteemi kõikidele osadele, samas kui tarnijad ja tellijad saavad täita oma vastavaid ülesandeid, sõltuvalt määratud õigustest.

**Turvalisus ja autentimine:** Turvalisuse tagamiseks kasutatakse kaasaegset autentimist (nt JWT tokenid) ja paroolide krüpteerimist (SHA-256). Kasutajate õigused määratakse nende rollide järgi, mis piirab juurdepääsu erinevatele süsteemi osadele.

**Tulemuslikkus ja usaldusväärsus:** Süsteem on optimeeritud kiireks laadimiseks ja suudab töödelda suure hulga päringuid ilma jõudluse kadumiseta. Kõik süsteemi vead on logitud ja administraatoritele on ette nähtud logide jälgimise ja analüüsi tööriistad.

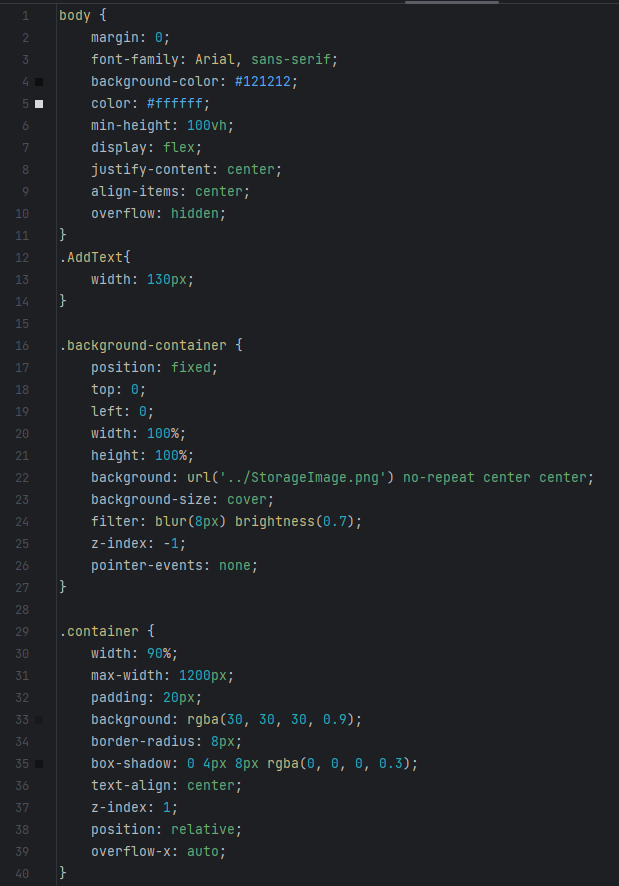
## Projekti ülevaade



# Praktiline osa

**Disaini arendamine, objektide joonistamine**

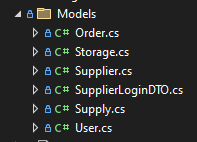
Eesmärk: luua intuitiivne, kohandatav ja atraktiivne kasutajaliides kõikidele rollidele.

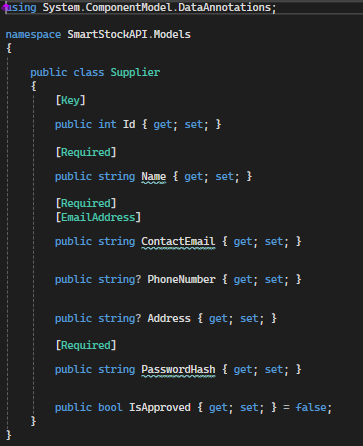


## Mudeli loomine

Eesmärk: Andmemudelite ja äriloogika väljatöötamine rakenduse jaoks.

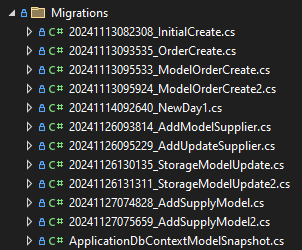
Näiteid mudelitest:

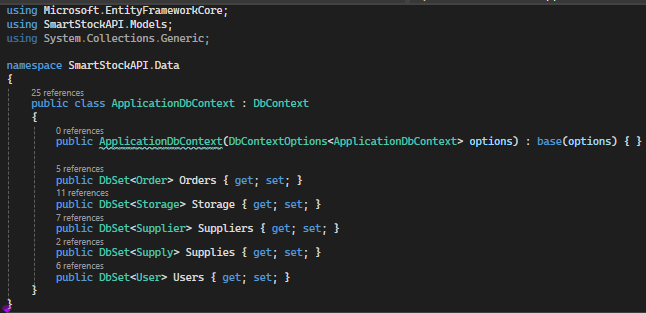
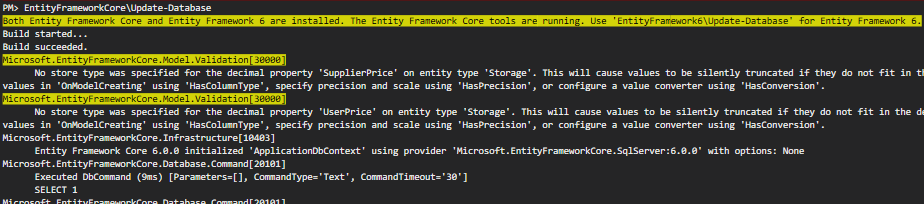




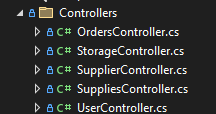
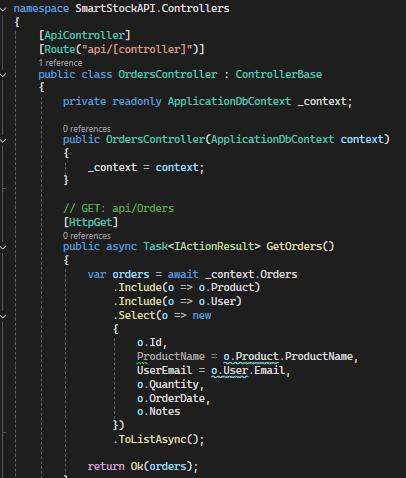
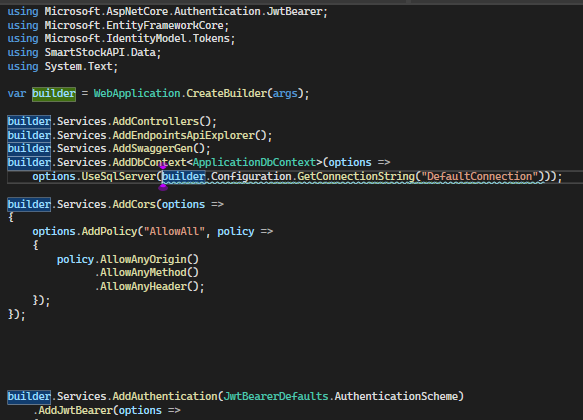
## Andmebaasi loomine

Eesmärk: projekteerida ja rakendada kohalik andmebaas kõigi rakenduse andmete säilitamiseks.



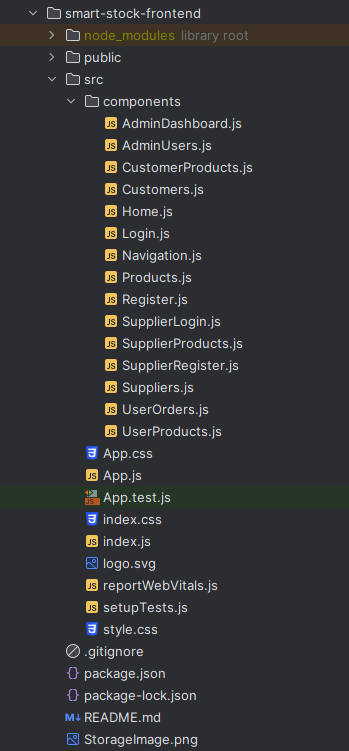
 

## Backend-i loomine

****** ****** 

## Frontend-i loomine

npx create-react-app smart-stock-frontend  
cd smart-stock-frontend  
npm i ja npm start



## Testimine ja silumine

Minu eesmärk on pakkuda kõrget kvaliteeti ja töökindlust. Selle saavutamiseks tegin põhjaliku testimise ja silumise, kasutades esmalt kohalikku serverit (localhost) erinevate testolukordade jaoks.

### Testimine Lokaalses Keskkonnas (localhost)

Testimise esialgne etapp toimus lokaalses keskkonnas, kasutades arenduskeskkonda, kus rakendus töötas otse mu arvutis.

**Lokaalses keskkonnas testimise eelised:**

* **Rakenduse funktsionaalsuse kontrollimine**   
  Lokaalses keskkonnas sain põhjalikult testida rakenduse funktsionaalsust, jälgides, kuidas kõik funktsioonid toimivad, ning veenduda, et rakendus käitub ootuspäraselt.
* **Vigade tuvastamine ja parandamine**   
  Kohene tagasiside lokaalsest keskkonnast võimaldas mul kiiresti tuvastada ja parandada kõik probleemid, suurendades rakenduse stabiilsust ja jõudlust.
* **Testimine arenduskeskkonnas**   
  Testides rakendust lokaalses serveris, suutsime luua erinevaid simulatsioone ja katsetada rakenduse käitumist erinevates olukordades, et tagada selle töökindlus.

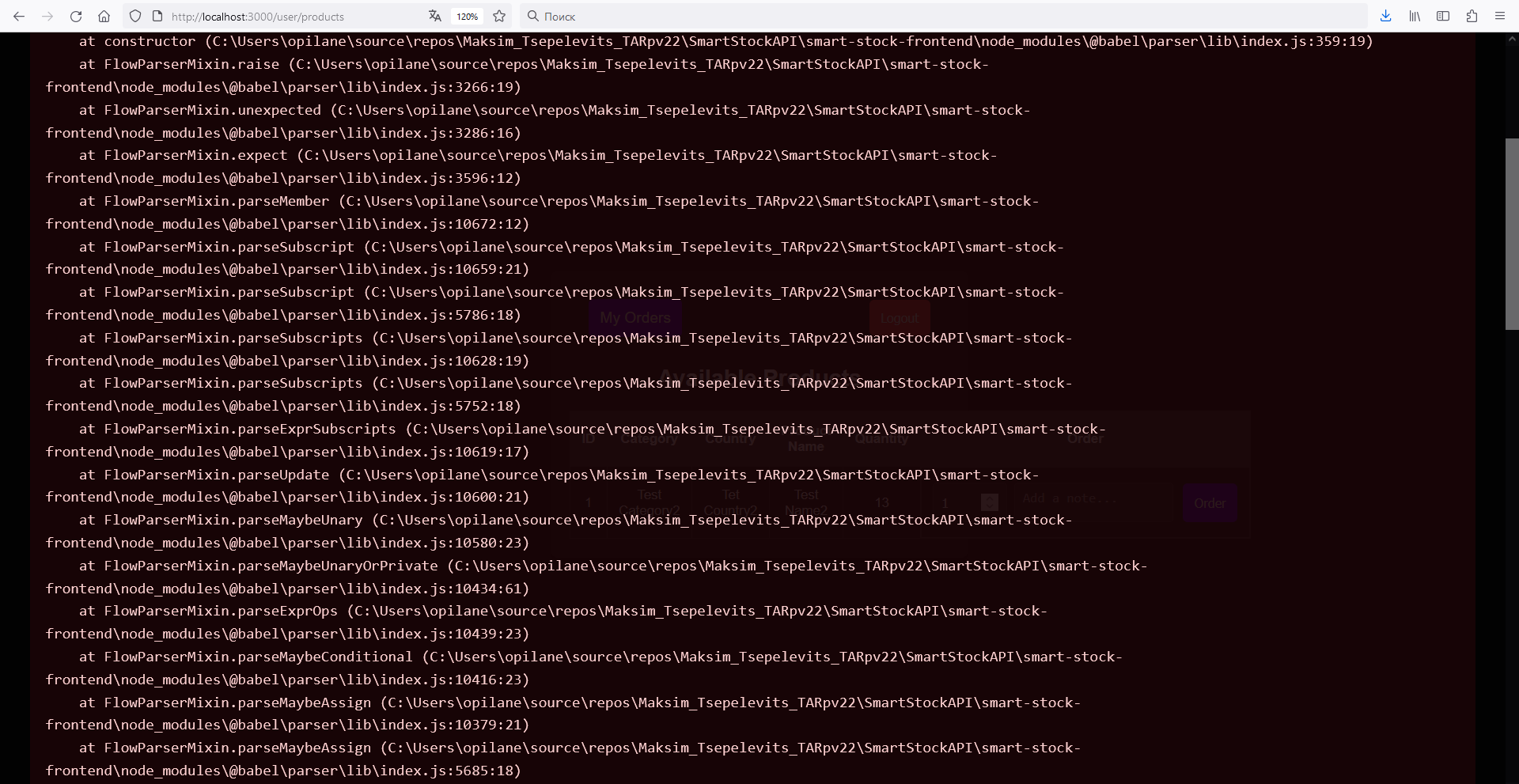
### Valmisoleku Testimine

Pärast edukat testimist lokaalses keskkonnas viisin läbi täiendava kontrolli, et veenduda rakenduse valmisolekus.

**Järgmised sammud:**

* **Manuaalne testimine lokaalselt**   
  Kontrollisin käsitsi iga funktsiooni ja kasutajaliidese aspekti, tagades, et kõik töötab nii, nagu peaks, ja kasutajakogemus on sujuv ja intuitiivne.
* **Jõudluse hindamine erinevates stsenaariumides**   
  Testisin rakendust erinevates situatsioonides ja seadistustes, et veenduda selle sujuvas ja stabiilses töös.

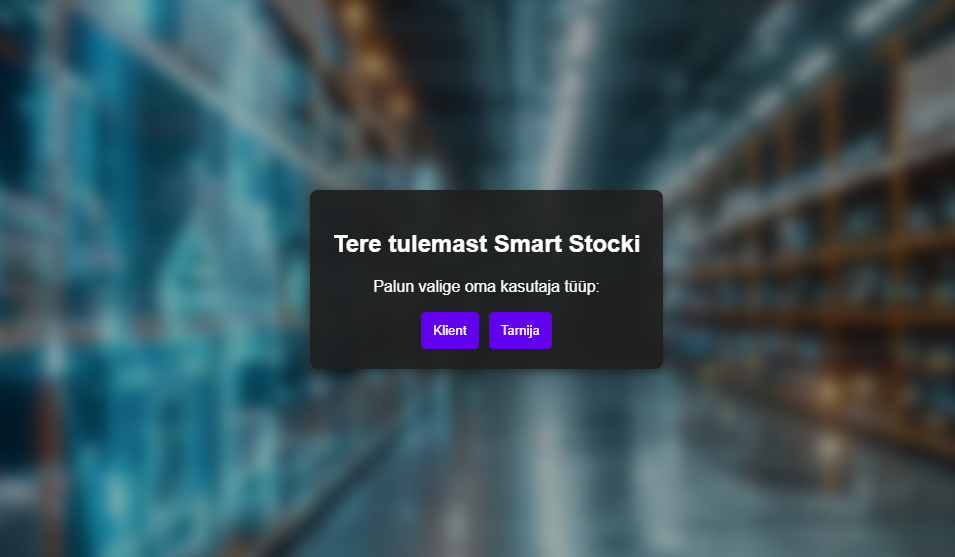
## Testimis kokkuvõte

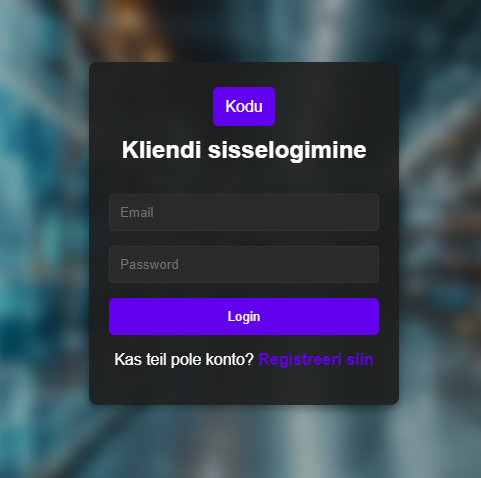
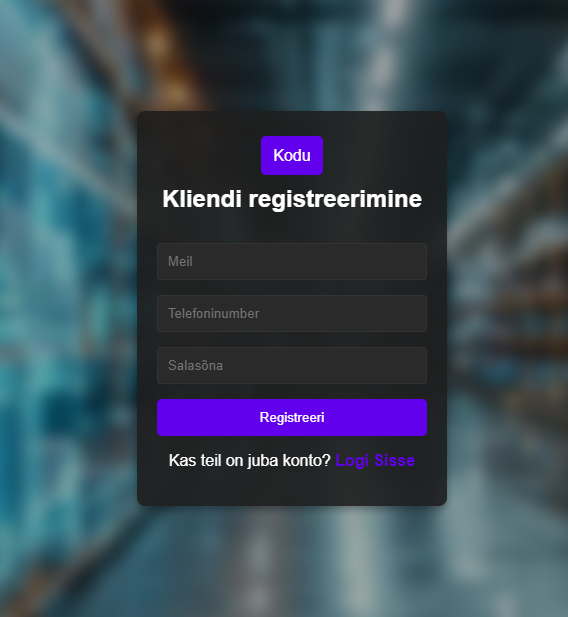
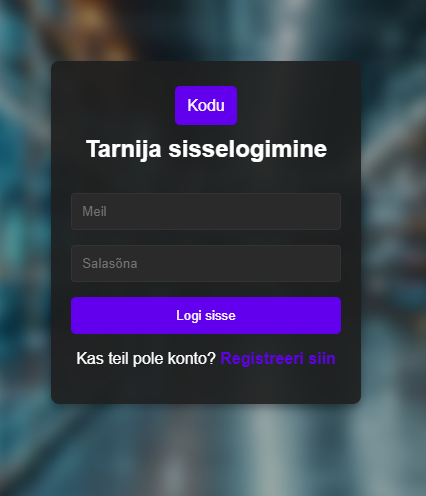
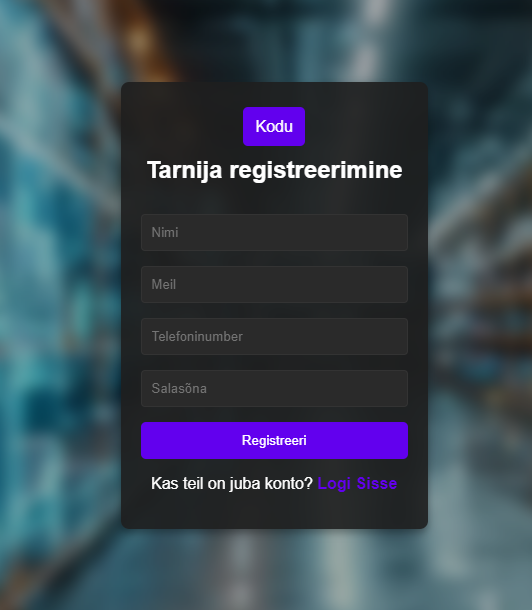
Tänu laiaulatuslikele emulaatorite ja pärisseadmete testimisele olen kindel, et laohaldussait on väga stabiilne, toimib ja seda on lihtne kasutada. Püüan tagada, et igal kasutajal oleks minu tootega parim kogemus ning jätkan selle täiustamist ja täiustamist andmete ja tagasiside põhjal.  
   


# Kasutusjuhend

## Pealeht

Kui alustate rakenduse avalehelt, näete mitmeid funktsioone. Esimene nupp kirjaga "Logi sisse kliendina" suunab teid sisselogimislehele, kus saate sisestada oma e-posti aadressi ja parooli, kui teil pole kontot, on registreerimiseks link, millel on vaja oma e-posti aadressi, telefoninumbri ja parooli sisestamiseks. Teine nupp kirjaga "Logi sisse tarnijana" viib teid tarnijate sisselogimislehele, kui teil pole kontot, on registreerimiseks link, kuhu peate sisestama oma nime, telefoninumbri , e-posti aadress ja parool. Kuid selleks, et saaksite pärast tarnijaks registreerumist sisse logida, on teil vaja administraatori kinnitust.

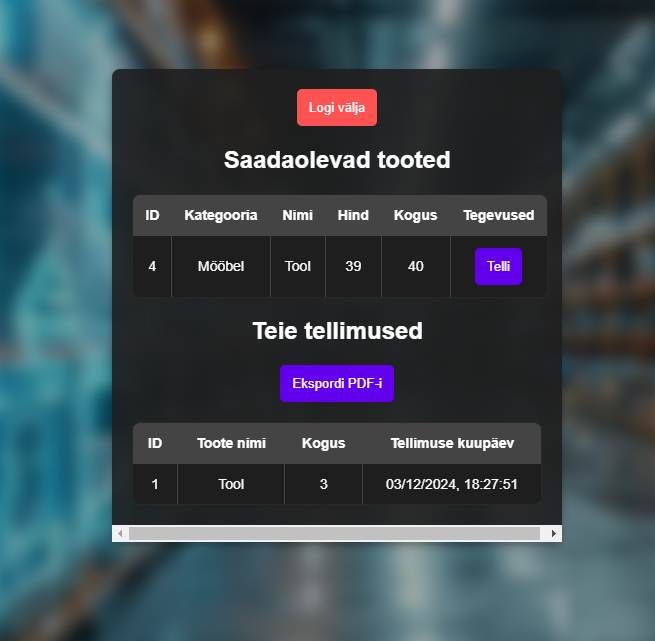
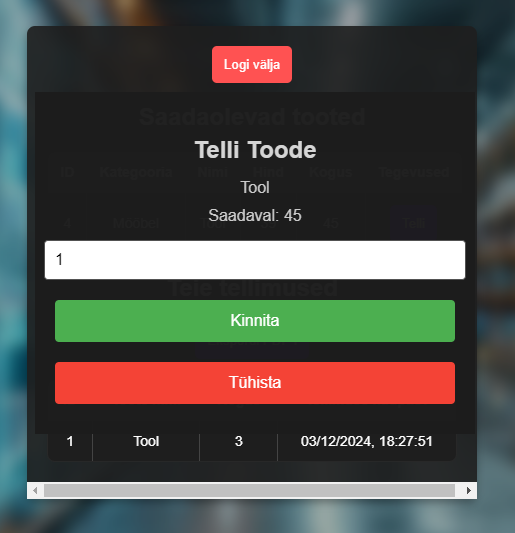


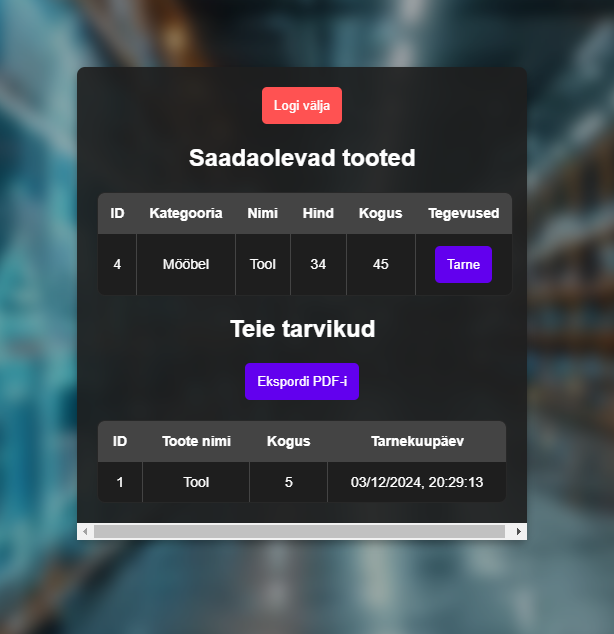
   

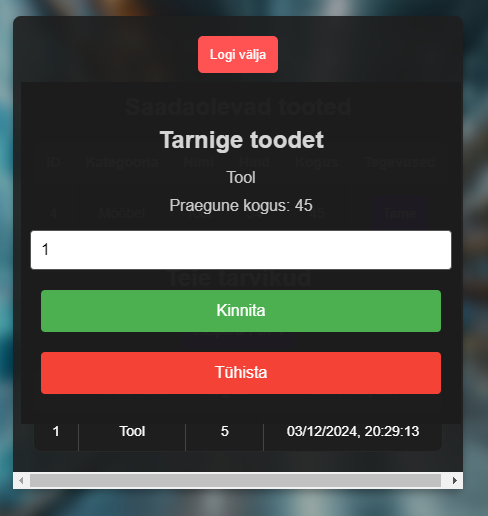
## Kastutaja poolt

Kui olete sisse loginud, kuvatakse sõltuvalt teie rollist erinevad valikud. Näiteks kui olete klient, on teil navigeerimisribal järgmised valikud: esimene nupp on "Logi välja", mis võimaldab teil oma kontolt välja logida; teine ​​nupp (toodete tabelis) on “Telli”, mis võimaldab esitada toodete tellimusi; kolmas nupp on “Ekspordi PDF-i”, mis võimaldab eksportida tellimuse andmed PDF-i.

Kui olete tarnija, siis pärast sisselogimist on teil navigeerimisribal järgmised valikud: esimene nupp on “Logi välja”, mis võimaldab kontolt välja logida; teine ​​nupp (toodete tabelis) on “tarne”, mis võimaldab esitada kauba tellimusi; kolmas nupp on “Ekspordi PDF-i”, mis võimaldab eksportida tellimuse andmed PDF-i.





## Administraatori poolt

**1. Logige sisse oma administraatori kontole.**

Administraatorina sisselogimiseks tuleb sisse logida kliendi meetodil. Sisestage oma administraatori e-posti aadress ja parool. Pärast edukat sisselogimist suunatakse teid administraatori juhtpaneelile.

**2. Administraatori juhtpaneel**

Pärast sisselogimist avaneb juhtpaneel järgmiste valikutega:

Väljalogimine: kui soovite administraatori kontolt välja logida, klõpsake nuppu Logi välja.

Kasutajahaldus: siin saate vaadata kasutajakontosid.

Tootehaldus: kõik tooted kuvatakse loendis. Siin saate tooteid lisada, redigeerida ja kustutada.

**3. Kasutajate haldamine**

Administraatorina saate:

Klientide haldamine: kõigi klientide vaatamiseks lehele sisenemiseks klõpsake nuppu „Kliendid”.

Tarnijate haldamine: klõpsake nuppu Tarnijad, et siseneda kõigi tarnijate kuvamise lehele, kus on võimalik tarnijaid kinnitada või tühistada, kuni tarnija on saanud kinnituse, et ta ei saa kontole sisse logida.

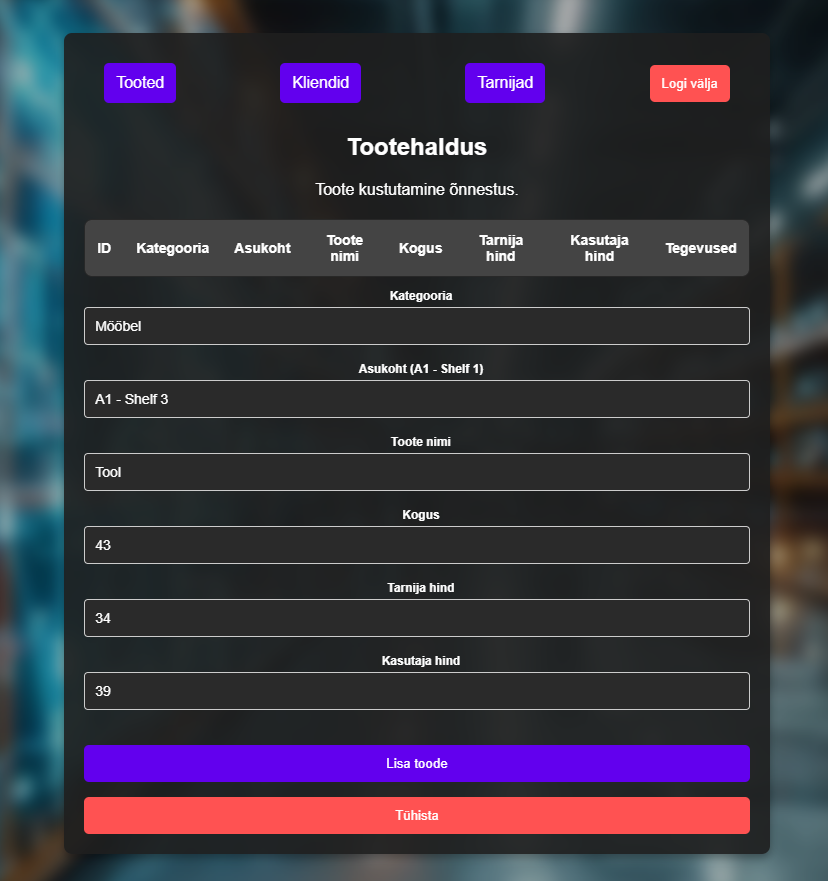
**4. Tootehaldus**

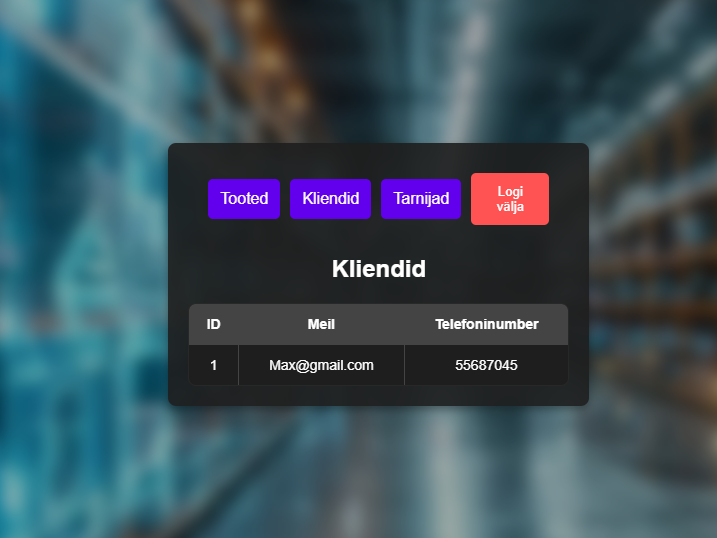
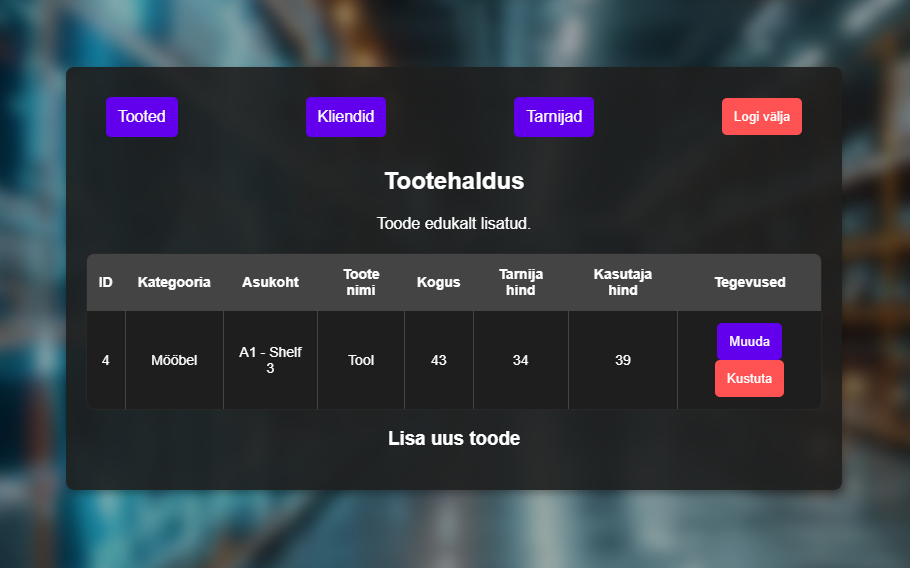
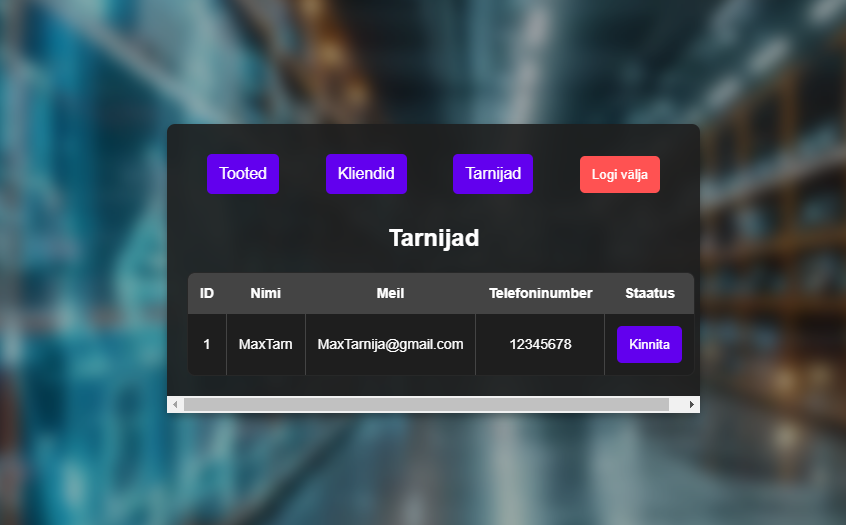
Administraatoril on võimalus:

Uute toodete lisamine: Vajuta nupule “Lisa toode”, sisesta toote kohta info (nimi, hind, kogus jne).

Toodete muutmine: valige loendist toode ja vajadusel muutke selle andmeid.

Toodete eemaldamine: Kui toodet pole enam vaja, valige loendist toode ja klõpsake nuppu "Kustuta".



# Kokkuvõte

**Mis on loodud:**

* Põhifunktsioonid kõikide kasutajarollide jaoks.
* Rakenduse põhifunktsionaalsus, mis põhineb kaupade ja laoseisu, tellimuste ja tarnete haldamisel.
* Töö MySQL andmebaasiga Entity Frameworki kaudu.
* React.js-i abil intuitiivne ja tundlik kasutajaliides.
* Põhilised turvafunktsioonid, sealhulgas kasutaja autentimine ja autoriseerimine.
* Toodete ja tarvikute lisamise, redigeerimise ja kustutamise ning tellimuste haldamise protsess.

**Mis ei õnnestunud:**

* Kõigi esialgsete plaanide täpne elluviimine õigeaegselt (tellimuste jälgimine ja olekuhaldus).
* Rakendage mõningaid mittefunktsionaalseid nõudeid, nagu logimine ja jälgimine.
* Adaptiivse liidese täielik rakendamine kõigi seadmete jaoks.
* Mõned probleemid andmete reaalajas sünkroonimisega.

**Täiendavad täpsustused:**

* Välisteenustega (maksesüsteemid ja teatised) integreerimise lõpuleviimine.
* Mittefunktsionaalsete aspektide (nt logimine, monitooring ja vigade käsitlemine) täiustamine.
* Funktsionaalsuse laiendamine, sealhulgas uute funktsioonide lisamine tarnijatele ja klientidele.
* Optimeerige jõudlust, et käsitleda rohkem kasutajaid ja tellimusi.
* Lahendatud on probleem mõne seadme liidese kohandatavusega.
* Andmete krüpteerimise ja lekete eest kaitsmise rakendamine.
* Regulaarse andmete varundamise rakendamine.

# Allikad

Microsoft. (2024). **ASP.NET MVC**. Allikas: ASP.NET MVC dokumentatsioon: <https://docs.microsoft.com/en-us/aspnet/mvc/>

Meta Platforms, I. (2024). **React.js**. Allikas: React.js dokumentatsioon: <https://reactjs.org/docs/getting-started.html>

Microsoft. (2024). **C#**. Allikas: C# dokumentatsioon: <https://learn.microsoft.com/en-us/dotnet/csharp/>

Microsoft. (2024). **Entity Framework**. Allikas: Entity Framework dokumentatsioon: <https://learn.microsoft.com/en-us/ef/>

Oracle Corporation. (2024). **MySQL**. Allikas: MySQL dokumentatsioon: <https://dev.mysql.com/doc/>

OWIN. (2024). **Open Web Interface for .NET (OWIN)**. Allikas: OWIN määratlus: <https://owin.org/>

Autofac. (2024). **Autofac**. Allikas: Autofac dokumentatsioon: <https://autofac.org/>

Microsoft. (2024). **Owin Integration**. Allikas: Owin ja Entity Frameworki integratsioon: <https://learn.microsoft.com/en-us/ef/core/providers/mysql/?tabs=dotnet-core-cli>

JWT.io. (2024). **JSON Web Tokens (JWT)**. Allikas: JWT dokumentatsioon: <https://jwt.io/introduction/>