

**Tallinna Tööstushariduskeskus
Noorem Tarkvaraarendaja eriala**

Aknarulode tootmine

PHP projekti kirjeldus

Juhendaja: Irina Merkulova

Õpilane: Pavel Ivanov

TARgv22

Tallinn 2023

Sisukord

Sisukord	2
1 Ülesanne	3
2 Töö plaan.....	4
2.1 Tehniline pool (kasutatav tarkvara)	5
3 Andmebaasiga töö	6
3.1 Andmetabelite loomine	6
4 Veebilehtede kirjeldused	7
4.1 registreeritote.php	7
4.2 riievalmis.php	9
4.3 puuvalmis .php.....	11
5 Klientrakenduse kasutajajuhend.....	15
5.1 Lisamine toote.....	15
5.2 vorm sisse logi-seks ja vorm registreerimiseks	16
5.3 Registreerimine vorm	17
5.4 4.7 lubamustrinr veebi leht	18
5.5 Otsing.....	22
5.6 lubamustrinr veebi leht kustutamine novaAlert.....	23
6 Kokkuvõte	24
7 Jooniste loetelu.....	25

1 Ülesanne

1. Koosta andmebaas peokuulutusega aknaruloode tootmine ja peale selle veebileht millega andmebaas võib seotatud.
2. Loo sinna juurde veebileht, kus saab tellimuse sisestada. Andmed talletatakse tabelisse (ei näidata veebilehel).
3. Loo SQL lause soovitud muustrinumbriga tellimuse lisamiseks. Lause määratud id-ga tellimuse riidelõikuse valmiks määramiseks. Lause valimis riide- ja puuosadega kuid veel pakkimata ruloode näitamiseks. Administraator saab vigaseid sisestusi ka kustutada
4. Loo tellimise tabel leht, et näha tellimused ja nende status kus kuvamine kas riie valmis või puu valmis.
5. Lisada ühine teemakohane kujundus kõikidele lehtedele eraldi css failiga.

2 Töö plaan

1. Saiidi päise navigeerimismenüü võimaldab juurdepääsu ülevaattetabelile, kus näeme kõiki lisatud tooteid ja millises valmistamise etapis need asuvad.
2. Uue tellimuse lisamise leht.
3. Pärast toote lisamist suuname lehele, kus on kuvatud kardinakanga valikud. Lehe funktsionaalsus võimaldab meil märkida kanga oleku "ei ole valmis," ja seejärel liigub toode "ei ole valmis" sektsiooni, kus ta jääb seni, kuni märgime selle "valmis."
4. Pärast "valmis" märkimist kardinakanga lehel suuname lehele, kus on esitatud ruloode valmistamiseks kasutatavate puidust detailide valikud. Siin saame märkida, kas toode on "valmis" või "ei ole valmis." Kui toode ei ole valmis, jääb see "ei ole valmis" tabelisse seni, kuni märgime, et see on valmis.
5. Veebi lehel on loodud registreerimine vorm milline on seotud andmebaasi tabeliga kasutajad sinna on registreeritakse kasutajad parooli räsiks on kasutatakse php muutuja (password_hash) nii siis veebilehel on võimalus logi sisse. Vaikevärtusele kasutajale enne registreerimist on lubatud näha ainult tellimuse lisamine.

Storyboard:

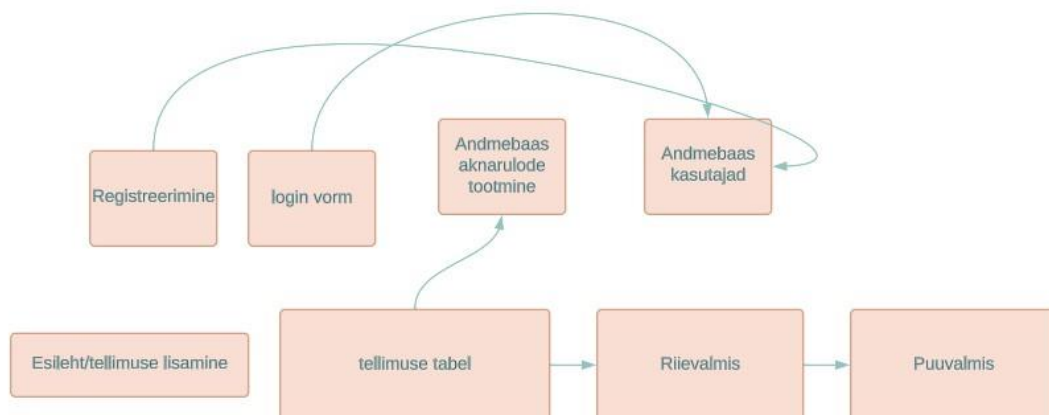


Figure 2-1Storyboard

2.1 Tehniline pool (kasutatav tarkvara)

Veebileht on loodud php versiooniga 5.6, HTML 5 ja CSS versioon 4. Nii siis on kasutanud SCSS, Haml, pug ning Javascript, dünaamiliste kujutiste ja kaasaegsete vormide loomiseks. Andmebaasiks on MariaDB ja rakenduse serveriks on zone.ee.

3 Andmebaasiga töö

3.1 Andmetabelite loomine

Localhostis lõin selle projekti jaoks... tabelid

1. See tabel Aknarul kajab Tabeli lehel kus kuvatakse status toote, kas riie,puu valmis või ei ole valmis ning kui riie ja puu valmis siis avaneb võimalus pakkida toote saatmiseks.

```
create table Aknarul(  
    id int primary key AUTO_INCREMENT,  
    mustri nr varchar(100),  
    riievalmis varchar(50),  
    puuvalmis varchar(100),  
    pakitud varchar (50)  
);
```

```
ALTER TABLE aknarul  
ADD COLUMN created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP;
```

id	mustri nr	riievalmis	puuvalmis	pakitud	luba	created_at
24	PÄIKESELINE TRIIBUTAJA	1	1	-1	1	2023-10-21 10:32:13
27	PORGANDINE ORANŽ	1	1	-1	1	2023-10-21 10:32:13
28	PORGANDINE VALGE	1	1	-1	1	2023-10-21 10:32:13
29	TUMEHALL RADIKAS	1	1	-1	1	2023-10-21 10:32:13
33	Mustri	1	1	-1	1	2023-10-21 10:47:31
124	Tttt	-1	-1	-1	-1	2023-11-01 14:18:50

id	mustri nr	riievalmis	puuvalmis	pakitud	luba
id : int(11)	mustri nr : varchar(100)	riievalmis : varchar(50)	puuvalmis : varchar(100)	pakitud : varchar(50)	luba : int(11)

Figure 3-1 Aknarulod table

2. kasutajad registreerimis- ja autentimisvormi jaoks. Kasutatakse password_hash mis bcrypt algoritm on tavaliselt 60 tähemärki.

```
create table kasutajad(  
  id int PRIMARY key AUTO_INCREMENT,  
  kasutaja varchar(50) unique,  
  parool varchar(60))
```

id	kasutaja	parool
17	Siil	\$2y\$10\$d80oTJhuizzEg.lpLw8DceiNZUCBKajBpKuA8u5rBzq...
18	Admin	\$2y\$10\$.6V6cQ5phu6TJia1Vcl6IOFr49p9qpn9GWxy/4DqKTJ...
29	Juhendaja	\$2y\$10\$oZWe/TJ0DYagYq0GFA6BH.3pmSJlyVwBFcsu3Lv5/IB...
31	Test	\$2y\$10\$AgRmLB56hVf7pl05DhMVU.cqmAGM4rqBXiqTrmuelUi...

id	kasutaja	parool
id : int(11)	kasutaja : varchar(50)	parool : varchar(60)

Figure 3-2Kasutajad table

4 Veebilehtede kirjeldused

4.1 registreeritote.php

Lisamine form

```
<!doctype html>  
<html lang="et">  
<head>  
  <title>Aknaruloode</title>  
</head>  
<body>  
<h1>Lisamine</h1>  
<form method="post">  
  <label for="uus_mustrinr">Lisa uus mustrinr:</label>
```

```
<input type="text" name="uus_muustrinr">
```

```
<input type="submit" value="Lisa">
```

```
</form>
```

```
</body>
```

```
</html>
```

UI tellimuse lisamiseks: tootesisend nihutatakse natuke paremale.



Figure 4-1UI tellimuse lisamiseks

Lisamiseks ridade on kasutatud selle php code...: Kui väli uus_muustrinr ei ole tühi, sisestatakse andmebaasi aknarul tabelisse uus väärtus (\$uus_muustrinr).


```

<?php
require_once("conf.php");
include('navigation.php');
global $yhendus;

if (!empty($_REQUEST["uus_muustrinr"])) {
    // Insert a new muustrinr into the aknarul table
    $uus_muustrinr = $_REQUEST["uus_muustrinr"];
    $kask = $yhendus->prepare("INSERT INTO aknarul (muustrinr) VALUES (?)");
    $kask->bind_param("s", $uus_muustrinr);
    $kask->execute();
}

?>

```

```

}

```

4.2 riievalmis.php

UI riievalmis kontroll kas valmis või mitte

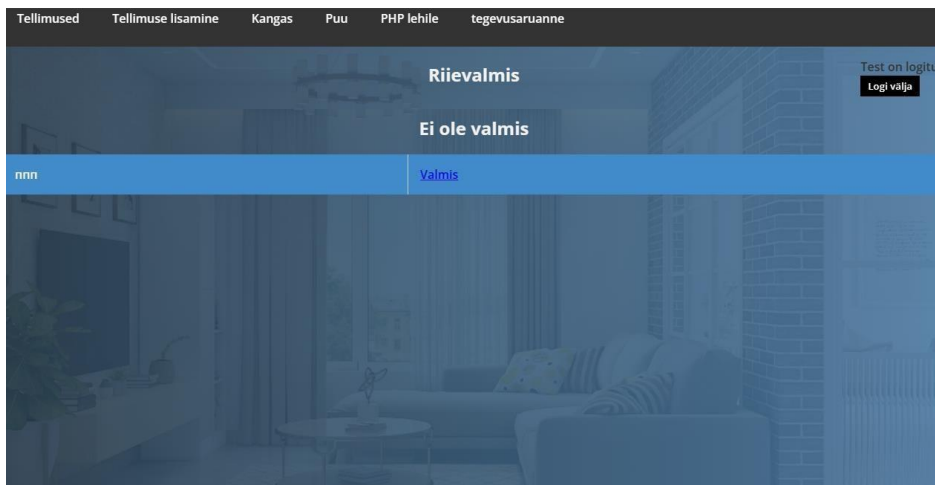


Figure 4-2UI riievalmis

PHP Code:

```

<?php
require_once("conf.php");
global $yhendus;

if (!empty($_REQUEST["riie_valmis_id"]) && !empty($_REQUEST["action"])) {
    $id = $_REQUEST["riie_valmis_id"];
    $action = $_REQUEST["action"];

    if ($action == "valmis") {
        $kask = $yhendus->prepare(
            "UPDATE aknarul SET riievalmis=1 WHERE id=?");
        $kask->bind_param("i", $id);
        $kask->execute();
        header("Location: slaloom.php");    exit;
    } elseif ($action == "eivalmis") {
        $kask = $yhendus->prepare(
            "UPDATE aknarul SET riievalmis=2 WHERE id=?");
        $kask->bind_param("i", $id);
        $kask->execute();
    }
}

$kask = $yhendus->prepare("SELECT id, muustrinr FROM aknarul WHERE (riievalmis=-1 )
");
$kask->bind_result($id, $muustrinr);
$kask->execute();
?>

```

See kood lubab meile kontrollida kas riie on valmis või mitte, me kasutame action kus 1 tähendab valmis, 2 ei ole valmis . Riievalmis=-1 tingimus, et kuvada kõik tellimused sellel lehel, sest meie disaini järgi oleme otsustanud kui riie ei valmis see jääb samal lehel aga läheb teisesse tabilisse mis nimi on ei ole valmis kus toote jääb enne tootja paneb seda valmiseks.

Tabel ei ole valmis riie:

```
<h1>Ei ole valmis</h1>
<table>
  <?php
    $kask_valmis = $yhendus->prepare("SELECT id, muustrinr FROM aknarul WHERE
riievalmis=2");
    $kask_valmis->bind_result($id, $muustrinr);
    $kask_valmis->execute();

    while ($kask_valmis->fetch())
  {      echo "          <tr>
          <td>$muustrinr</td>
          <td><a href='?riie_valmis_id=$id&action=valmis'>Valmis</a></td>
          </tr>
          ";
    }
  ?>
</table>
```

Ei ole valmis =2, seetõtu tigmusel me kajastame just sellel tabelis ainult need rulod mis ei valmis veel, aga samal ajal millalgi riie peaks olema valmis siis sellel juhul meie tabelis ei ole valmis on üks link valmis, mis peale vajatumist muudab riie valmiseks ja muudab status ei ole valmis meie tabelis valmiseks

4.3 puuvalmis.php

PHP Code puuvalmis:

```
<?php if (isset($_GET['code'])) {die(highlight_file(__File__, 1)); }?>
```

```

<?php require_once("conf.php"); global $yhendus; if
(!empty($_REQUEST["puuvalmis_valmis_id"]) && !empty($_REQUEST["action"])) {
    $id = $_REQUEST["puuvalmis_valmis_id"];
    $action = $_REQUEST["action"];

    if ($action == "valmis") {
        $kask = $yhendus->prepare(
            "UPDATE aknarul SET puuvalmis=1 WHERE id=?");
        $kask->bind_param("i", $id);
        $kask->execute();
    } elseif ($action == "eivalmis") {

        $kask = $yhendus->prepare(
            "UPDATE aknarul SET puuvalmis=2 WHERE id=?");
        $kask->bind_param("i", $id);
        $kask->execute();
    }
}
$kask = $yhendus->prepare("SELECT id, muustrinr FROM aknarul WHERE (puuvalmis=1 )
");
$kask->bind_result($id, $muustrinr);
$kask->execute();
?>

```

"See kood võimaldab meil kontrollida, kas riie on valmis või mitte. Kasutame toimingut, kus 1 tähendab "valmis" ja 2 tähendab "ei ole valmis". Kui "Riievalmis" on -1, siis see tingimus lubab meil kuvada kõik tellimused sellesse tabelisse. Meie kujunduse järgi oleme otsustanud, et kui riie ei ole valmis, jääb see samale lehele, kuid kantakse teise tabelisse, mille nimi on "Ei ole valmis".

Tabel puuvalmis:

```

<h1>Puuvalmis</h1>
<table>

```

```

<?php while ($kask-
>fetch()) { echo "
<tr>
<td>$mustrinr</td>
<td><a href='?puuvalmis_valmis_id=$id&action=valmis'>Valmis</a></td>
<td><a href='?puuvalmis_valmis_id=$id&action=eivalmis'>Ei ole valmis</a></td>
</tr>
";
}
?>
</table>

```

UI Tabel puuvalmis:

Puuvalmis		
Mustr	Valmis	Ei ole valmis
test	Valmis	Ei ole valmis
hk	Valmis	Ei ole valmis

Figure 4-3UI puuvalmis

Tabel puu ei ole valmis:

```

<table>
  <?php
    $kask = $yhendus->prepare("SELECT id, muustrinr FROM aknarul WHERE
puuvalmis=2");
    $kask->bind_result($id, $muustrinr);
    $kask->execute();

    while ($kask->fetch())
  {      echo "      <tr>

          <td>$muustrinr</td>
          <td><a href='?riie_valmis_id=$id&action=valmis'>Valmis</a></td>
        </tr>
";
    }
  ?>
</table>

```

UI tabel puu ei ole valmis:

ei ole valmis	
test	Valmis
hk	Valmis

"Ei ole valmis = 2, seetõttu tingimusel, et me kajastame selles tabelis ainult need rulood, mis ei ole veel valmis. Kuid samal ajal, kui riie peaks olema valmis, siis sel juhul meie tabelis "Ei ole valmis" on üks link "Valmis," mis pärast vajutamist muudab riide valmiseks muudab tingimused tabelis.

5 Klientrakenduse kasutajajuhend

Meie projekt hetkel ei luba pakkuda kasutajatele laia valikut funktsioonide veebi lehel.

5.1 Lisamine toote

Hetkel on loogiliselt me võime pakkuda tavalisile kasutajatele ainult üks vebileht toote lisamine.



uue toote lisamisel ilmub novaAlert, et kasutaja kinnitab oma tellimuse või loobub.

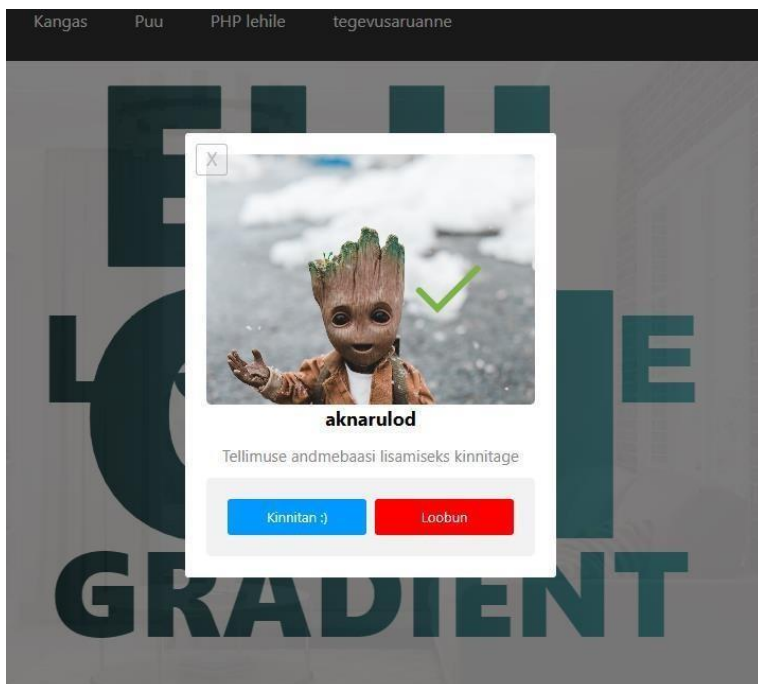


Figure 5-1NovaAlert

tausta loomiseks oleme kasutanud css style sellest veebilehest: [text gradient](https://codepen.io) (codepen.io)

meie veebileht on loodud mitu vorme millised on seotud andmebaasi tabeliga.

5.2 vorm sisse logi-seks ja vorm registreerimiseks

Oleme teinud log sisse vorm scss ja pug abil.

1. Kõige pealt on tuleb tuvastada globaalselt npm install pug
2. Peale selle saame projekti kasti cmd abil ning tuvastame seal veel kord npm install pug, ja lisaks sellele npm install pug-cli -g, pärast seda võime muuda meie pug file: pug -P loginform.pug.

```
Found 0 vulnerabilities

C:\Users\Admin\Desktop\Aknaruloode_tootmine>pug -P loginform.pug
'pug' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Admin\Desktop\Aknaruloode_tootmine>npm install pug-cli -g
npm WARN deprecated core-js@2.6.12: core-js@<3.23.3 is no longer maintained and not recommended for usage due
Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown up to 100x
lled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of
added 84 packages in 13s

13 packages are looking for funding
  run `npm fund` for details

C:\Users\Admin\Desktop\Aknaruloode_tootmine>pug -P loginform.pug
  rendered loginform.html

C:\Users\Admin\Desktop\Aknaruloode_tootmine>
```


Loginform.pug

Reegistreemise vormiks oleme kasutanud scss ning html.pug ja scss sellest alikast: [Finance Mobile Application-UX/UI Design Screen One \(codepen.io\)](#)

Tullemus:

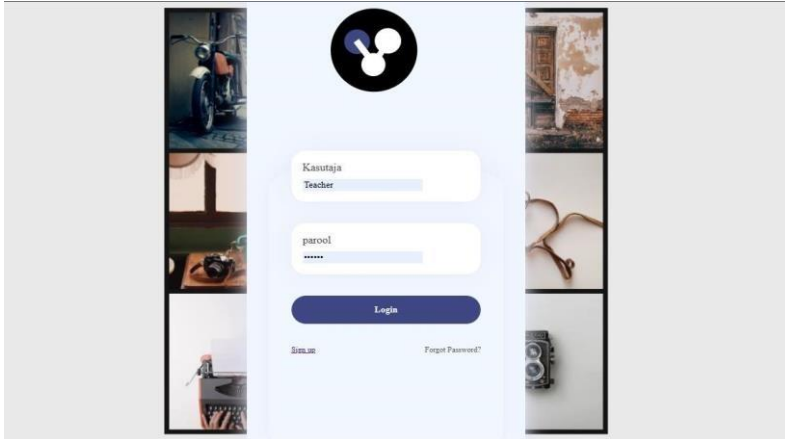


Figure 5-2 Login form

5.3 Registreerimine vorm

Reegistreemise vormiks oleme kasutanud scss ning html.Haml sellest alikast: [CSS Cube Login Form \(codepen.io\)](#)

Konverteerimiseks oleme kasutanud veebi leht: [Haml to ERb/HTML online converter – haml2erb.org](#). See vorm on seotud php sql käskidega milised lubavad meile regesreerida inimesi otse kasutajate tabelisse. Muutujatena kasutatakse name kasutaja ning parool.

Input [Haml](#)

```
.bg-top
.bg-inner
.bg-right
.bg-inner
.bg
.bg-inner
.text
  log In
.credits
%a{ :href => "https://codepen.io/marko-zub/", :target => "_blank" } (line 6, column 74):
```

Convert

Output [HTML \(with ERb\)](#)

```
<form autocomplete="off" class="form">
  <div class="control">
    <h1>
      Sign In
    </h1>
  </div>
  <div class="control block-cube block-input">
    <input name="username" placeholder="Username" type="text"></input>
  <div class="bg-top">
    <div class="bg-inner"></div>
  </div>
```

Tulemus:



Figure 5-3 Sign up form

See sign in vorm võib peeldada natuke ebakorektselt edge Brauseris, kuigi, näiteks Mozilla brauseris pole vigu aga see vormi funktsionaalsust ei mõjuta.

5.4 4.7 lubamustrinr veebi leht

Lubamustrinr veebi lehel on lisatud tabel scss abil. Oleme ksautanud selle alikat selleks sihiks: [Sass: Table Styling \(codepen.io\)](#) konvertimiseks oleme kasutanud command promt. Näide:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3448]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd desktop
C:\Users\Admin\Desktop>cd Aknaruloode_tootmine
C:\Users\Admin\Desktop\Aknaruloode_tootmine>sass table.scss table.css
C:\Users\Admin\Desktop\Aknaruloode_tootmine>_
```

Kastis väljapääsu juures oleme saanud selle css:

Css:

```
body { background-color: #efefef; }

.table-latitude { width: 100%; }

.table-latitude caption { text-transform: uppercase; font-weight: bold; color: #fab700;
text-align: center; } .table-latitude thead { border-top: 2px solid #fab700; border-
bottom: 2px solid #fab700; }

.table-latitude thead th { text-transform: uppercase;
```

```

    text-align: center;
padding: 10px;    color:
#006ac6;    border: 1px solid
#F1F1F1; }
.table-latitude tbody tr {    borderbottom:
1px solid #F1F1F1;    color:
#75787b; }
.table-latitude tbody tr td {
padding: 10px;    border: 1px
solid #F1F1F1; } .tablelatitude
tbody tr th {    padding-left:
10px;    border:
1px solid #F1F1F1; }
.table-latitude tfoot {    color:
#fab700;    font-style: italic;    border-
bottom: 2px solid #fab700; }
.table-latitude tfoot tr td {
padding: 10px;    border: 1px solid
#F1F1F1; }

/*# sourceMappingURL=table.css.map */

```

Muudasime vastavalt html ja php osad:

```

<link rel="stylesheet" type="text/css" href="table.css">
<form action=""
method="post">    </form>

<div class="container">
    <div class="panel panel-default">
        <div class="panel-heading">
            <h1></h1>
        </div>
        <div class="panel-body">
            <table class="table-latitude">
<caption>tooteteave</caption>
            <thead>

```

```

        <th>mustrinr</th>
        <th>rievalmis</th>
        <th>puuvalmis</th>
        <th>lloodud</th>
        <th>pakkida</th>

    </thead>
    <tfoot>
    <tr>
        <td colspan=3>Contact details for the Knowledge and Development
Team</td>
    </tr>
</tfoot> <?php while ($kask->fetch()) {
    $asendatud_rievalmis = asenda($rievalmis);
    $asendatud_puuvalmis = asenda($puuvalmis);
    $loalahter = ".";
    if ($luba == 1) {
        $loalahter = "saadetud";
    }
    if ($luba == -1 && $puuvalmis == 1) {
        $loalahter = "<a href='?vormistamine_id=$id'>saatma</a>";
    }
    echo "

    <tr>
        <td>$mustrinr</td>
        <td>$asendatud_rievalmis</td>
        <td>$asendatud_puuvalmis</td>
        <td>$created_at</td>
        <td>$loalahter</td>
        <td><a href='?kustuta=$id'>Kustuta</a></td>
    </tr>
";

}
$kask->close();    if
(isset($_REQUEST["kustuta"])) {
    $kask = $yhendus->prepare("DELETE FROM aknarul WHERE id=?");
    $kask->bind_param("i", $_REQUEST["kustuta"]);
    $kask->execute();
}
?> </table>

```

Tullemus:

Tabel Tellimuse lisamine Kangas Puu Tagasi PHP lehile

Teacher on logitud
Logi välja

Tellimused

TOOTETEAVE

MUSTRINR	RIEVALMIS	PUUVALMIS	LOODUD	PAKKIDA
PÄIKESELINE TRIIBUTAJA	valmis	valmis	2023-10-21 10:32:13	saadetud Kustuta
PORGANDINE ORANŽ	valmis	valmis	2023-10-21 10:32:13	saadetud Kustuta
PORGANDINE VALGE	valmis	valmis	2023-10-21 10:32:13	saatma Kustuta
TUMEHALL RADIKAS	valmis	valmis	2023-10-21 10:32:13	saadetud Kustuta
Mustri	valmis	valmis	2023-10-21 10:47:31	saadetud Kustuta
test	valmis	valmis	2023-10-26 11:57:27	saatma Kustuta

Contact details for the Knowledge and Development Team

Figure 5-4 Tellimused table

5.5 Otsing

Otsingu bar loomiseks oleme kasutanud selle alikat: [Inspiration Search Bar \(codepen.io\)](https://codepen.io/InspirationSearchBar) koosneb scss ja html-st.

Tullemus:

1.



5.6 lubamustrinr veebi leht kustutamine novaAlert

Kustutamiseks tellimuse ilmub animated alert mis teavitab meid et record on kustutatud andmebaasist. Code on kasutanud sellest alikast: [Simple Alerts \(codepen.io\)](https://codepen.io)



The screenshot shows a web application interface with a search bar and a table of orders. A large red overlay with the text 'REKORD KUSTUTATUD' is positioned over the table. The table has columns for 'MUSTRINR', 'RIE', 'LOOTETAV', 'LOODUD', and 'PAKKIDA'. The rows list various items like 'PÄIKESELINE TRIIBUTAJA', 'PORGANDINE ORANŽ', 'PORGANDINE VALGE', 'TUMEHALL RADIKAS', 'Mustri', 'test', 'mmn', and 'POOLPIMENDAV'.

MUSTRINR	RIE	LOOTETAV	LOODUD	PAKKIDA
PÄIKESELINE TRIIBUTAJA	valmis		2023-10-21 10:32:13	saadetud Kustuta
PORGANDINE ORANŽ	valmis		2023-10-21 10:32:13	saadetud Kustuta
PORGANDINE VALGE	valmis	valmis	2023-10-21 10:32:13	saadetud Kustuta
TUMEHALL RADIKAS	valmis	valmis	2023-10-21 10:32:13	saadetud Kustuta
Mustri	valmis	valmis	2023-10-21 10:47:31	saadetud Kustuta
test	valmis	valmis	2023-10-26 11:57:27	saatma Kustuta
mmn	ei valmis	.	2023-10-27 10:43:56	. Kustuta
POOLPIMENDAV	.	.	2023-10-29 10:10:35	. Kustuta

Figure 5-5 Simple alert

Selleks oleme kasutanud css mis kood määrab stiilid visuaalselt atraktiivse animeeritud hoiatus-/teatiskasti loomiseks. Hoiatuskast sisaldab animeeritud elemente, nagu kasvav joon ja tekst, mis liiguvad ja ilmuvad visuaalselt meeldival viisil. Animatsiooni juhivad CSS-i võtmekaadrid. Kast on algselt peidetud ja muutub käivitamisel nähtavaks.

6 Kokkuvõte

Mis on tehtud?

1. Andmebaasi loomine aknaruloode tootmise jaoks ning sellele seotud veebilehe loomine.
2. Veebileht, kus saab sisestada tellimusi, ning andmete talletamine tabelisse (kuid tellimusi ei näidata veebilehel).
3. SQL-lause loomine tellimuse lisamiseks soovitud muustrinumbriga, tellimuse riidelõikuse valmiks määramiseks ning tellimuste kuvamiseks riide- ja puuosadega, kuid veel pakkimata ruloodega. Lisaks on võimalus administraatoril vigaseid sisestusi kustutada.
4. Tellimuste tabeli leht, kus saab näha tellimusi ja nende olekut (kas riie on valmis või puu on valmis).
5. Ühine kujundus kõigile lehtedele eraldi CSS-failiga.

Mida saab veel arendada antud töös?

1. Aruannete loomine: funktsionaalsus aruannete ja statistika genereerimiseks tellimuste, toodete või klientide kohta. See aitab paremini jälgida ettevõtte tegevust ja tulemusi.
2. Kasutajasõbralikud teatised: kasutajatele sõbralikud teated ja veateated, et nad saaksid aru, millised toimingud on edukad ja kui midagi läheb valesti.
3. Kasutajate autentimine ja õiguste haldamine: kasutajate autentimine ja rollid, mis võimaldavad erinevatel kasutajagruppidel (nt administraatorid ja tavalised kasutajad) teatud tegevusi teha või piiravad juurdepääsu teatud funktsioonidele.

Kokkuvõttes võib edasine arendamine keskenduda veebirakenduse funktsionaalsuse, turvalisuse, kasutatavuse ja jõudluse parandamisele, et tagada parim võimalik kasutajakogemus ja ettevõtte tõhus tegevus.

7 Jooniste loetelu

Figure 2-1Storyboard.....	4
Figure 3-1Aknarulod table.....	6
Figure 3-2Kasutajad table.....	7
Figure 4-1UI telimuse lisamiseks	8
Figure 4-2UI riievalmis	9
Figure 4-3UI puuvalmis.....	13
Figure 5-1NovaAlert.....	16
Figure 5-2Login form	17
Figure 5-3Sign up form	18
Figure 5-4Tellimused table.....	22
Figure 5-5Simple alert	23