# OPPIMISPORTAALIN KÄYTTÖLIITTYMÄN TOTEUTUS

LAB-AMMATTIKORKEAKOULU
Tieto- ja viestintätekniikan insinööri AMK
Tietojenkäsittely
Kevät 2021
Niko Juntunen

**Tiivistelmä**

| Tekijä(t) | Julkaisun laji | Valmistumisaika |
|---|---|---|
| Juntunen, Niko | Opinnäytetyö, AMK | Kevät 2021 |
| | Sivumäärä | |
| | 29 | |

Työn nimi

**Oppimisportaalin käyttöliittymän toteutus**

Tutkinto

Tieto- ja viestintätekniikka, ohjelmistotekniikka, Insinööri AMK


Tiivistelmä

Tämän opinnäytetyön tavoitteena oli toteuttaa oppimisportaalin frontend ASP.NET Core Razor pages-ohjelmistokehyksellä. Tavoitteena oli saada aikaan pilottiversio Piilaakso Academy Oy:lle. Tavoitteisiin kuului myös CSS- ja JavaScript-kirjastojen hyödyntäminen projektissa. Oppimisportaalin käyttäjät jaetaan eri rooleihin tarpeen mukaan. Portaalin avulla käyttäjä voi kehittää omaa osaamistaan ja seurata sitä. Taidoilla ja sertifikaateilla käyttäjän kehityksestä tulee konkreettisempaa ja näkyvämpää.

Razor pages on ASP.NET Core:n lisäosana toimiva www-projektien kehittämiseen suunniteltu ohjelmistokehys. Siinä käytetään MVVM-mallia (Model-View-ViewModel). Razor pages mahdollistaa C#-kielen käytön suoraan HTML-sivulla. Bootstrap mahdollistaa sivuston responsiivisuuden ja siitä on hyötykäytetty muitakin osia mm. kortteja (cards). JQueryn-kirjastoa tarvitaan suurimmaksi osaksi muiden kirjastojen toimivuuden vuoksi. Font Awesome toi ikoneiden avulla eloa ulkoasuun.

Sovellus perustuu palvelimella olevaan tietokantaan, joka pitää kirjaa käyttäjän tiedoista ja taidoista. Sovellukseen kirjaudutaan tunnuksilla sisään. Käyttäjän etusivuna toimii hallintapaneeli, jota kautta käyttäjä pääsee nopeasti hallitsemaan omia tietojaan. Sovelluksessa on myös useita eri haku mahdollisuuksia, jotta etsiminen olisi helpompaa.

Opinnäytetyön tavoitteisiin päästiin ja pilottiversio saatiin valmiiksi. Lähes kaikki ominaisuudet saatiin toimiviksi, mutta myös lisättäviä ominaisuuksia jäi. Oppimisportaalin kehitys jatkuu vielä ja ongelmien ratkaiseminen on seuraava askel Piilaakson toimesta.



Asiasanat

oppimisportaali, frontend, CSS, JavaScript, responsiivinen, Razor pages, Model-View-ViewModel, HTML, C#

## Abstract


| Author(s) | Type of publication | Published |
|---|---|---|
| Juntunen, Niko | Bachelor's thesis | Spring 2021 |
| | Number of pages | |
| | 29 | |


Title of publication

**Interface implementation of Employee Learning Portal**

Name of Degree

Bachelor of Engineering, Information and Communication Technology


Abstract

Purpose of this thesis was to create an Employee learning portal's frontend using ASP.NET Core with Razor pages. The main goal was to create pilot version of the product to Piilaakso Academy Oy. Other objectives were to utilize CSS and JavaScript libraries in the project. Employee learning portal users are divided into different roles as needed. The portal allows the user to develop and monitor their own skills. User development becomes more concrete and visible with skills and certificates.

Razor pages is an ASP.NET Core extension for web project development. It uses MVVM-model (Model-View-ViewModel). Razor pages allows you to use C#-language directly on an HTML page. With Boostrap it is easy to create the site to be responsive and it gives other good toolsets to be used, like cards. JQuery library are mostly needed for the functionality of other libraries. Font Awesome icons brought the layout to life.

The application is based on a database on the server that keeps track of the user skills. You log in to the application with credentials. The users frontpage is a dashboard where the user can quickly control their own information. The application also has several different search options to make finding things easier.

The goals of the thesis were achieved and the pilot version was completed. Almost all features were made functional but there were also features to be added. The development of the Employee learning portal is still ongoing and solving these problems is the next step by Piilaakso.



Keywords

Employee learning portal, frontend, CSS, JavaScript, Responsive, Razor pages, Model-View-ViewModel, HTML, C#

SISÄLLYS

# 1 JOHDANTO

Työn tekeminen ja oppiminen ovat korona-aikana kokeneet isoja muutoksia. Koronan vuoksi on siirrytty työskentelemään ja opiskelemaan etänä, varsinkin teknologia-aloilla. Nykyisin myös verkossa käytävät kurssit ja materiaalit auttavat itsensä kehittämistä nopeasti ja tehokkaasti. Itsensä kehittämistä ja oppimista on hyödyllistä myös seurata jo senkin vuoksi, että konkreettisesti pystyy näkemään oman kehityksensä. Uusien asioiden oppiminen ja vanhojen taitojen kehitys auttavat sinua ja työnantajaa. Niiden avulla voidaan seurata, että tarvitseeko jokin taito lisää kehitystä vai löytyykö jo riittävä osaaminen.

Piilaakso Academy on vuonna 2018 perustettu digitaalisen osaamisen kehittämiseen ja työllistymisen edistämiseen erikoistunut yritys. Piilaakso Academy sai alkunsa vallitsevasta osaajapulasta, joka koettelee erityisesti ohjelmisto- ja digialaa. Piilaakso Academy tukee ja sparraa yrityksiä digiosaamisen kehittämisessä ja lupaa löytää ja kouluttaa sopivat tekijät erilaisiin teknologia tarpeisiin. (Piilaakso Academy, 2021.)

Tämän opinnäytetyön tavoitteena on luoda oppimisportaalin pilottiversio käyttäen ASP.NET Core 3.1:tä ja Razor pages-tekniikkaa. Lisäksi tavoitteena on saada käyttöliittymästä moderni ja käyttäjäystävällinen. Tämä opinnäytetyö keskittyy projektin frontend-toteutukseen.

Tässä opinnäytetyössä teoria-osuudessa käydään läpi ensin oppimisportaalin roolituksia, taitoja ja kursseja. Sen jälkeen siirrytään käymään läpi eri CSS- ja JavaScript-kirjastoja, joita projektissa on hyödynnetty. JQuery, Bootstrap, Font Awesome, sekä muut kirjastot ovat iso osa kokonaisuutta, joiden avulla saadaan sivusta modernimman näköinen, paremman näköinen ja responsiivinen. Microsoft Razor pages-tekniikan avulla C#-ohjelmointi on mahdollista suoraan HTML:ssä. Käytännön osuudessa käydään oppimisportaalin käyttöliittymää yleisesti läpi ja mitä sen on tarkoitus tehdä ja mitä käyttäjä niistä hyötyy.

# 2 OPPIMISPORTAALI

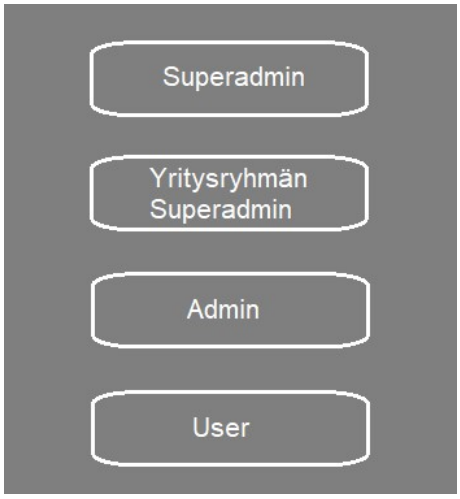
## 2.1 Yleistä

Oppimisportaalin tulee olla selaimessa toimiva sovellus, jossa pidetään ylhäällä käyttäjän tietoja, taitoja, kursseja, oppitunteja, yrityksiä ja ryhmiä. Sen kautta täytyy pystyä seuraamaan käyttäjän kehittymistä ja mahdollistaa se, että käyttäjä voi myös kehittää itseään.

Oppimisportaalissa kehitystä täytyy pystyä seuraamaan erilaisten taitojen ja sertifikaattien avulla. Taitojen seuranta tulee tapahtua numeroiden avulla (0-5) ja erilaisia sertifikaatteja täytyy pystyä myöntämään suoritetuista kursseista.

## 2.2 Roolit

Oppimisportaalissa tulee olla superylläpitäjiä (superadmin), Yritysryhmän superylläpitäjiä, ylläpitäjiä (admin) ja käyttäjiä (user). Näistä sovelluksen ylläpitäjä on superadmin, jolla on eniten oikeuksia tehdä muutoksia sovelluksessa. Toiseksi eniten oikeuksia on Yritysryhmän superylläpitäjillä. Kolmanneksi eniten oikeuksia on adminilla ja vähiten käyttäjällä (kuva 1). Yritysryhmän superylläpitäjä on superylläpitäjän ja ylläpitäjän väliin sijoittuva rooli, jonka tarkoituksena on ylläpitää yritysryhmiin kuuluvia töitä.



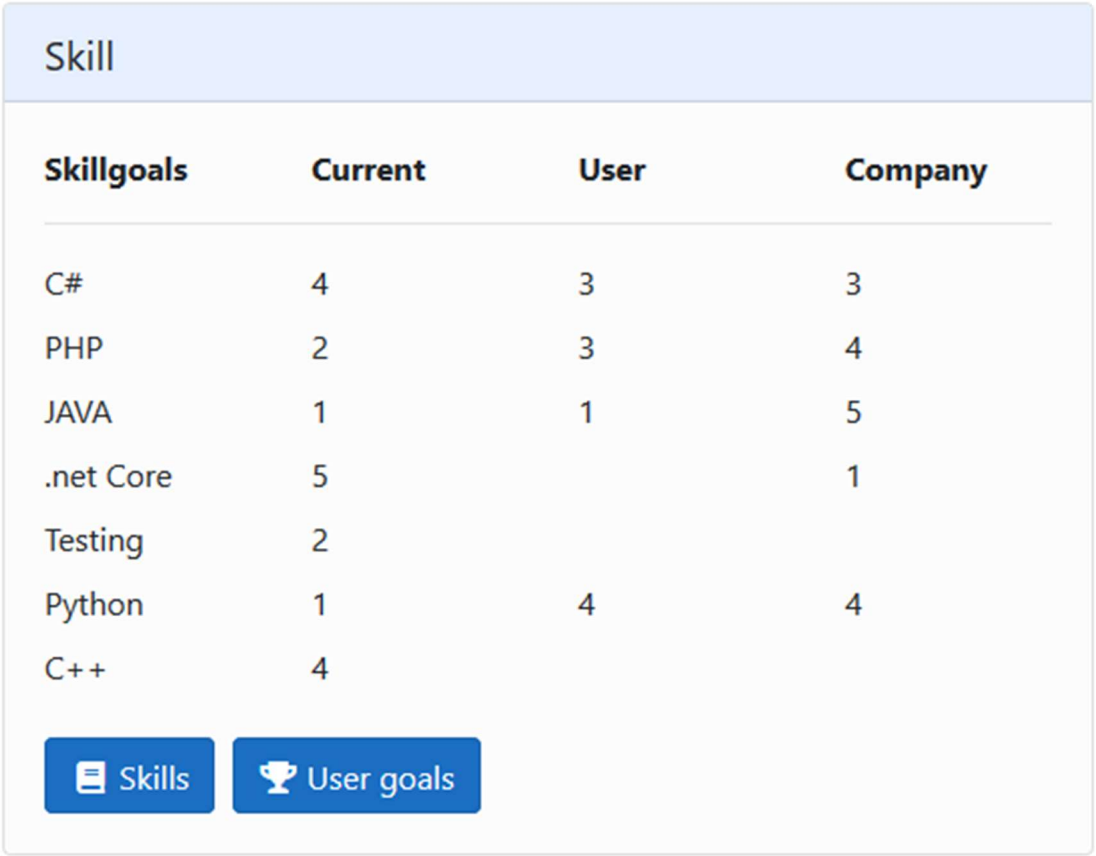

Kuva 1. Oppimisportaalin rooli-järjestys

Superadminit pystyvät muokkaamaan kaikkia käyttäjiä ja luomaan uusia yrityksiä ja yritysryhmiä. Superadminien tarkoitus on olla sovelluksen ylin taso ja pystyä muokkaamaan kaikkea. Erikseen on vielä yritysryhmäkohtaiset superadminit, jotka eivät voi muokata, kuin yritysryhmien sisällä olevia asioita. Adminit voivat luoda yrityksen sisällä uusia ryhmiä ja hallita niiden käyttäjiä. Adminit ovat yrityskohtaisia. Käyttäjät voivat vain

tutkia omia tietojaan ja kehitystään. Käyttäjien oikeudet ovat suppeat, eivätkä he voi muokata kuin omia tietojaan.

## 2.3 Taidot ja sertifikaatit

Oppimisportaalin tulee olla mahdollisimman toimiva ratkaisu käyttäjälle ja yritykselle, jotta molemmat voivat seurata käyttäjän kehitystä. Sen tulee mahdollistaa käyttäjälle erilaisten kurssien käyminen oppimisportaalissa, joiden avulla käyttäjän ja yrityksen on helpompi kehittää käyttäjää. Käyttäjän tiedot lisätää SQL-tietokantaan, josta tiedot haetaan näytettäväksi. Kehitystä seurataan numeroin (0-5). Käyttäjän, yrityksen ja ryhmien täytyy pystyä määrittelemään omia tavotteitaan (kuva 2).




Kuva 2. Taitotason seuranta

Käyttäjä ja työnantaja hyötyvät molemmat näistä tiedoista, jotta työntekijää voidaan kehittää oikein. Tiedot helpottavat myös itse käyttäjää, koska siitä on helppo seurata, mitkä taidot vaativat lisäkoulutusta ja mitkä ovat jo tarpeeksi hyvällä tasolla. Mikäli käyttäjällä ei ole ollenkaa arviota taidosta, on sen numeron tilalla tyhjä ruutu.

Sertifikaatteja voidaan myöntää kurssien suorittamisista. Mikäli käyttäjällä on jo entuudestaan olemassa oleva sertifikaatti tietystä kurssista, pystyy hän pyytämään

sovelluksen ylläpitäjää hyväksyttämään sen itselleen. Sertifikaatit näkyvät käyttäjälle profiili-sivulla.

## 2.4 Kurssit

Ohjelman kautta tulee olla mahdollista myös käydä kursseja, jotka auttavat työntekijän osaamisen kehittämistä tarpeen vaatiessa. Kursseja tulee pystyä selaamaan läpi ja valitsemaan itselleen sopivimman ajankohdan ja sijainnin perusteella. Kurssin tiettyjä oppintuntejakin on mahdollista valita, mikäli kurssin pitäjä on niitä erikseen määritellyt.

Kurssien HTML-puolella pitää pystyä kutsumaan Modelia, joka hakee tiedot tietokannasta controllerin avulla ja tämän jälkeen tulostaa tiedot sivulle (kuva 3).

```
58  @foreach (var item in Model)
59  {
60      <tr>
61          <td>
62              @Html.DisplayFor(modelItem => item.FirstName)
63          </td>
64          <td>
65              @Html.DisplayFor(modelItem => item.LastName)
66          </td>
67          <td>
68              @Html.DisplayFor(modelItem => item.Email)
69          </td>
70
71          <td>
72              @Html.DisplayFor(modelItem => item.PhoneNumber)
73          </td>
74          <td>
75              @Html.DisplayFor(modelItem => item.EmpStatus)
76          </td>
77
78          }
```

Kuva 3. Kurssin HTML

Superadmin näkee kaikki kurssille osallistuvat käyttäjät ja pystyy myös tätä kautta seuraamaan, ketkä kurssin ovat suorittaneet. Tätä kautta kurssin pitäjä antaa myös arvosanan kurssin päätyttyä. Mikäli kurssin on joutunut jättämään kesken, tai sille ei ole muusta syystä päässyt, näkyy sekin tieto tätä kautta superadminille, sekä kurssin pitäjälle.

# 3 KÄYTETYT KIRJASTOT

## 3.1 JQuery

JQuery on ilmainen avoimen lähdekoodin JavaScript-kirjasto, joka julkaistiin vuonna 2006. Se tekee HTML-dokumentin manipuloinnista, tapahtumien käsittelystä, animaatioista ja Ajaxin käytöstä yksinkertaista ja se tukee useimpia selaimia. JQuery on nopea, monipuolinen ja helposti laajennettavissa oleva kirjasto. Seuraavissa kuvissa on esitetty DOM-puun solmujen valintaa selectorin avulla, sekä niiden manipulointia ja tapahtuman käsittelyä. (Jquery, 2021.) DOM esimerkissä haetaan button-elementti, jonka luokkana on ”continue” ja vaihdetaan sen HTML-osuus ”Next Step...” (kuva 4).

```
1 | $( "button.continue" ).html( "Next Step..." )
```

Kuva 4. DOM esimerkki

Tapahtuman käsittelyn esimerkissä tuodaan esille #banner-message -elementti, joka on aluksi näkymätön display:none CSS-määritelmän vuoksi. Mutta kun klikataan missä vain #button-container -elementin kohdassa tulee se esille (kuva 5).

```
1 | var hiddenBox = $( "#banner-message" );
2 | $( "#button-container button" ).on( "click", function( event ) {
3 |   hiddenBox.show();
4 | });
```

Kuva 5. Tapahtuman käsittelyn esimerkki

JQueryä hyväksikäyttävässä JavaScript ohjelmassa on vähemmän lähdekielistä ohjelmaa, kuin normaalisti JavaScriptiä käytettäessä HTML-sivulla. JQueryn dokumentaatio on myös laadukasta ja kattavaa. JQuery on niin suosittu, että siihen löytyy myös helposti valmiita esimerkkejä ja liitännäisiä.

JQuery-liitännäiset ja muut JavaScript-tiedostot, jotka käyttävät jQuery-kirjastoa, täytyy sijoittaa ohjelmakoodiin jQuery-kirjaston jälkeen (kuva 6).

```
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/select2@4.1.0-beta.1/dist/js/select2.min.js">
</script>
<script
type="text/javascript" src="~/lib/bootstrap/dist/js/bootstrap.bundle.js">
</script>
```

Kuva 6. JQuery sijoitus

JQuery ja siihen liittyvät projektit on julkaistu MIT-lisenssin alla. Tämä tarkoittaa sitä, että tekijänoikeustiedot säilytetään muokkaamattomana, jolloin jQuery on käytettävissä missä projektissa tahansa (myös kaupallisissa).

## 3.2 Bootstrap

Bootstrap on ilmainen avoimen lähdekoodin CSS-kirjasto, jota käytetään usein tekemään sivuista responsiiviset ja käyttäjäystävällisemmät. Bootstrap julkaistiin 19.08.2011 ja tällä hetkellä se on maailman suosituin avoimen lähdekoodin front-endin työkalupakki, joka sisältää Sass-muuttujia ja sekoituksia, responsiivisen ruudukko-järjestelmän, useita eri komponentteja, sekä tehokkaat JavaScript-laajennukset. Bootstrap myös hyödyntää jQuerya, mutta Bootstrap-kirjastoa voi käyttää myös pelkästään CSS-kirjastona, jolloin jQuerya ei tarvita. (Bootstrap, 2021.)

Bootstrapin CSS-kirjasto otetaan käyttöön laittamalla seuraava HTML-määrittely (kuva 7), HTML:n head-tagien sisään ylimmäksi. Bootstrapin JavaScript-kirjasto tulee jQuery-kirjastohaun alle, jotta molemmat kirjastot toimivat oikein sivulla (kuva 8).

```
<head>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" runat="server" />
</head>
```

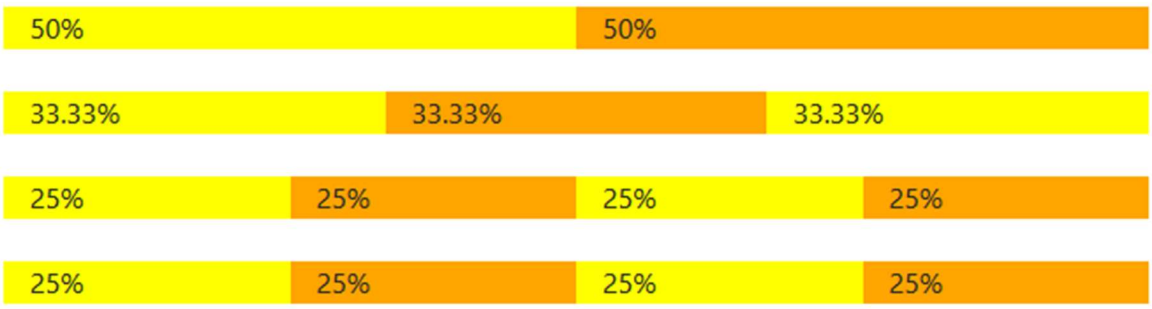
Kuva 7. Bootstrapin CSS-kirjasto HTML-sivulla

```
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script
type="text/javascript" src="~/lib/bootstrap/dist/js/bootstrap.bundle.js">
</script>
<script type="text/javascript"
        src="~/lib/bootstrap-select-1.13.14/dist/js/bootstrap-select.min.js">
</script>
```

Kuva 8. Bootstrapin JavaScript-kirjasto HTML-sivulla

Bootstrapin tarkoitus on nopeuttaa ja helpottaa kehittäjien työtä ja siihen se tarjoaa laajan valikoiman valmiita, helposti kutsuttavia funktioita ja tyylimäärittelyjä. Näihin lukeutuu erikokoisille näytöille ja laitteille valmiit määrittelyt, jotta dynaamisten ja skaalautuvien sivujen luonti olisi helpompaa.

Bootstrapin ruudukko (grid) ominaisuuden ansiosta sivu skaalautuu saumattomasti työpöytäkoosta mobiilikokoon, joten sivu on suoraan käytettävissä myös mobiilialustoilla. Ruudukkoa apuna käyttäen on sivuston rakenteen luominen helpompaa, koska sen avulla siitä saadaan symmetrisen näköinen, sekä sisältö, että sen hallinta helpottuvat. Seuraavassa kuvassa nähdään Bootstrapin ruudukosta, miten sitä voidaan jakaa ja miten se toimii (kuva 9, kuva 10). Kuvassa 9 on ensin jaettu sivu kahteen osaan, sitten kolmeen osaan ja viimeiset pari riviä neljään osaan. Kuvassa 10 nähdään kun ruudun koko pienenee alkaa Bootstrapin ruudukon responsiivisuus toimia ja kahteen osaan jaettu ruudukko menee allekkain. Kolmeen osaan jaettu menee myös allekkain ja neljään osaan jaettua voidaan käyttää joko allekkain tai vierekkäin.




Kuva 9. Bootstrapin ruudukko ominaisuus


Kuva 10. Bootstrapin ruudukko kun näyttö on alle 576 pikseliä leveä

Bootstrapin kortit (cards) ominaisuus on joustava sisältösäiliö. Se sisältää vaihtoehdot otsikoille, alaviitteille ja sisällölle. Näiden lisäksi on useita eri mahdollisuuksia asiayhteyteen sisältyviin taustaväreihin ja tehokkaisiin näyttövaihtoehtoihin. Kortteja voi käyttää myös useampaa kerralla ja niitä voidaan käyttää allekkain tai vierekkäin (kuva 11).


Kuva 11. Kortteja

Korttien käyttäminen on yksinkertaista ja niihin voi sisällyttää esimerkiksi kuvia tai painonapin avulla linkin muualle. Kortit ovat myös responsiivisia, joten ne muokkautuvat näytön koon mukaan automaattisesti. Korttien ollessa vierekkäin ja näytön koon pienentyessä tarpeeksi pieneksi, menevät kortit automaattisesti allekkain. Korttien koko määräytyy tekstin mukaan, mutta vierekkäin olevia kortteja voi myös määritellä niin, että alaviittaukset ovat samalla tasolla. Kun alaviitteet ovat samalla tasolla keskenään, määritellään korttien koko sen mukaan, missä on eniten tekstiä.

Korteissa on myös mahdollista käyttää navigointia. Navigoinnin avulla korttia voi käyttää usealla eri tavalla. Siihen voidaan lisätä joko suoraan linkki uudelle sivulle tai mahdollisesti

toteuttaa se niin, että kun kortin navigoinnista valitaan toinen ikkuna, vaihtaa se kortin sisältöä.

## 3.3 Font Awesome

Font Awesome on ikoni-kirjasto ja työkalu, joka on suunniteltu käytettäväksi inline-elementtien kanssa. Inline-elementti toimii siten, ettei se vaihda riviä. Font Awesomissa käytetään HTML:ssä i-tagia, mutta myös span-tagissa sen käyttö on mahdollista (kuva 12).

```
<i class="fas fa-camera"></i> <!-- this icon's 1) style prefix == fas and 2) icon name == camera -->
<i class="fas fa-camera"></i> <!-- using an <i> element to reference the icon -->
<span class="fas fa-camera"></span> <!-- using a <span> element to reference the icon -->
```

Kuva 12. Font Awesomen ikonin käyttöä HTML:ssä

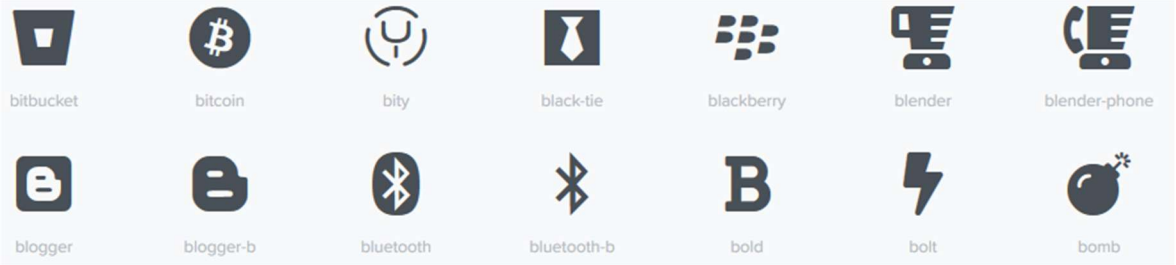
Font Awesomen ikonit perivät automaattisesti CSS-määrittelyn koon ja värin, mikä tarkoittaa, että ne muovautuvat tekstin mukana mihin ikinä niitä sijoitetaankaan (kuva 13).

```
<span style="font-size: 3em; color: Tomato;">
  <i class="fas fa-camera"></i>
</span>

<span style="font-size: 48px; color: Dodgerblue;">
  <i class="fas fa-camera"></i>
</span>

<span style="font-size: 3rem;">
  <span style="color: Mediumslateblue;">
  <i class="fas fa-camera"></i>
  </span>
</span>
```

Kuva 13. Ikonin perintä ja käyttö

Font Awesome sisältää jo yli 7 tuhatta ikonia ja määrä kasvaa kokoajan. Osa näistä ikoneista on tosin maksullisia, mutta myös ilmaisia löytyy. Yhdelle ikonille voi löytyä myös useampi eri vaihtoehto, kuten esimerkiksi käänteiset värit tai vähän parempi laatu.

Yleensä näistä vaihtoehdoista ensimmäinen vaihtoehto ilmainen ja loput vaihtoehdot kuuluvat Font Awesomen maksulliseen versioon (kuva 14). (Font Awesome, 2021.)



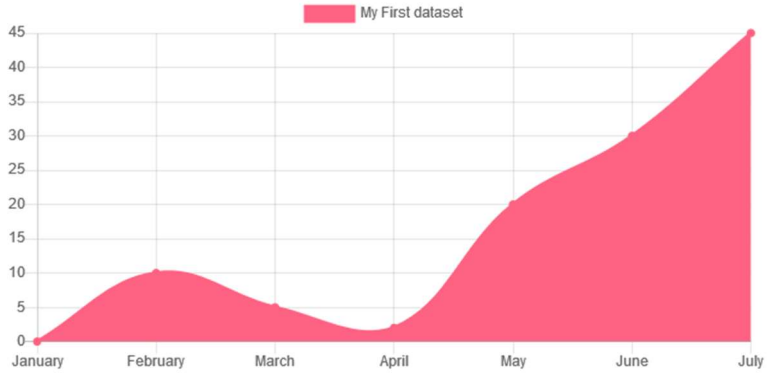

Kuva 14. Font Awesomen ilmaisia ikoneita

## 3.4 Chartjs

Chartjs on ilmainen kaavioiden luomiseen tarkoitettu työkalu, jota on mahdollisuus muokata omien tarpeiden mukaiseksi. Se on JavaScriptillä toimiva monipuolinen kaavioiden luontiin tarkoitettu työkalu. Sen dokumentaatio on selkeää ja responsiivisuuden ansiosta se sopii myös hyvin mobiilialustoille. Chartjs on helppokäyttöinen ja se otetaan HTML-sivulla käyttöön linkittämällä script-tagien sisään sen kutsu, jonka jälkeen se määritellään HTML-sivulle canvas-tagien sisään.

Luodaan viivakaavio ”myChart” id:llä, jonka leveys ja korkeus on 400. Kaavion luomisen jälkeen siihen päästään käsiksi JavaScriptillä juuri tuon ”myChart”-id:n kautta, jonka jälkeen kaavio määritellään JavaScriptissä mieluisaksi. Esimerkissä luodaan viivakaavio JavaScriptin puolella, joka sisältää kuukaudet tammikuusta – heinäkuuhun ja johon on syötetty valmiiksi tietoja 0, 10, 5, 2, 20, 30 ja 45 (kuva 15). (Chartjs, 2021.)

```
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
<body>
    <canvas id="myChart"></canvas>
</body>
<script>
var ctx = document.getElementById('myChart').getContext('2d');
    var chart = new Chart(ctx, {
        // The type of chart we want to create
        type: 'line',

        // The data for our dataset
        data: {
            labels: ['January', 'February', 'March',
             'April', 'May', 'June', 'July'],
            datasets: [{
                label: 'My First dataset',
                backgroundColor: 'rgb(255, 99, 132)',
                borderColor: 'rgb(255, 99, 132)',
                data: [0, 10, 5, 2, 20, 30, 45]
            }]
        },

        // Configuration options go here
        options: {}
    });</script>
</html>
```




Kuva 15. Viivakaavio

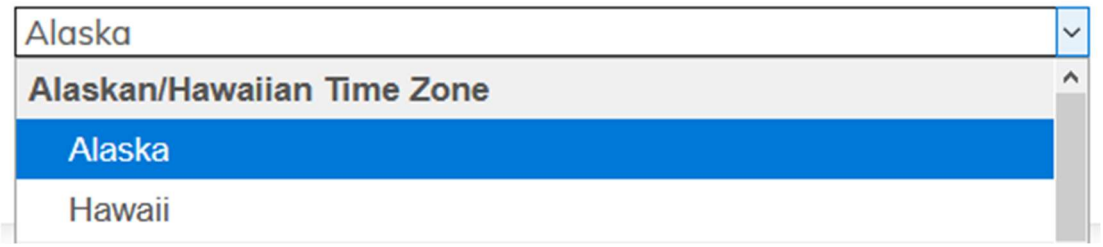
Chartjs:n käyttö soveltuu parhaiten tilanteisiin, missä tiedetään entuudestaan halutut kaaviotyypit. Sillä mikäli tarvitaan runsas määrä lisäominaisuuksia ja asiakkaan täytyy pystyä muokkaamaan kaavioita niin halutessaan, voi jostain muusta kaaviokirjastosta saada paremman hyödyn.

Chartjs tarjoaa useita erilaisia kaavioita, mistä voi valita itselleen sopivimman. Yhdestä eri kaavioista voi olla myös erilaisia vaihtoehtoja. Esimerkiksi pylväskaavioista löytyy pystykaavio, pinottu pylväskaavio ja viivakaavio. Kaavioita pystyy myös animoimaan niin halutessaan. Chartjs tarjoaa kattavan määrän esimerkkejä, joista on helppo lähteä rakentamaan omanlaisensa kaavio.

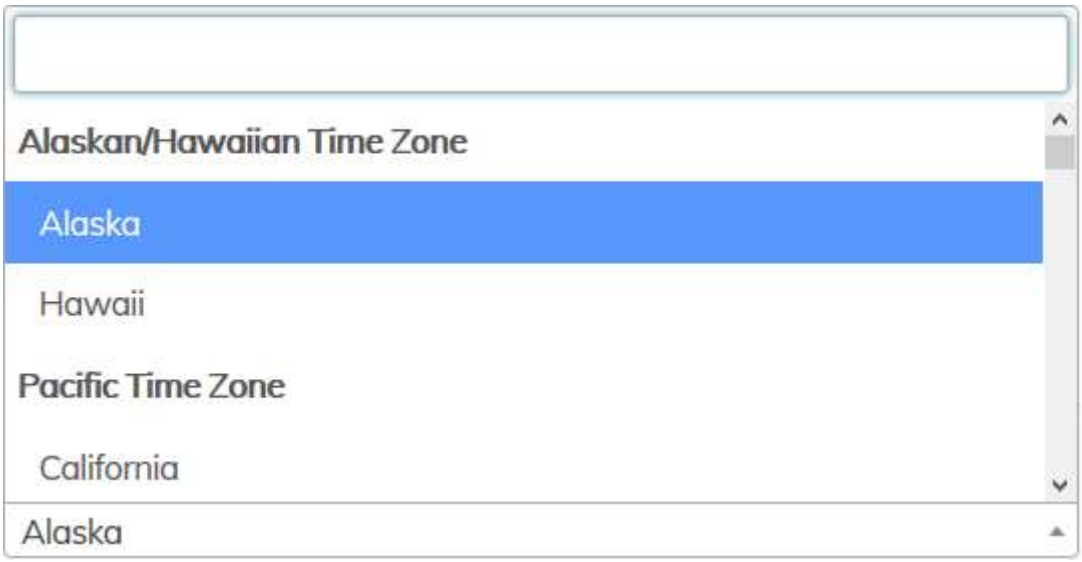
## 3.5 Select2

Select2 on tarkoitettu korvaamaan jQueryn valinta-laatikot. Select2 antaa muokattavissa olevan valinta-laatikon, johon mukana tulee mahdollisuus hakea ja merkitä. Mukaan tulee

myös loputon vierittäminen ja monia muita usein käytettyjä ominaisuuksia. Select2 otetaan käyttöön HTML-sivulla linkittämällä se head-tagien sisään. Select2:n avulla on helppo muokata normaalista valinta-laatikosta (kuva 16) nykyaikaisemman näköinen (kuva 17). (Select2, 2021.)




Kuva 16. Normaali jQueryn valinta-laatikko




Kuva 17. Select2:lla muokattu valinta-laatikko

Select2 rekisteröi itsensä jQuery funktioksi, jonka vuoksi sitä voidaan kutsua missä vain jQueryn selectorissa, johon select2 halutaan tilalle. Ensin täytyy varmistaa, että DOM on valmis muokattavaksi, jonka jälkeen select2:ta voidaan kutsua (kuva 18).

```
// In your Javascript (external .js resource or <script> tag)
$(document).ready(function() {
    $('.js-example-basic-single').select2();
});
```

Kuva 18. Select2 kutsu

Select2 valinnat rakennetaan HTML-sivulle select-tagien sisään, jolloin sille voidaan luoda erilaisia vaihtoehtoja valinta-laatikkoon. (kuva 19).

```
<select class="js-example-basic-single" name="state">
  <option value="AL">Alabama</option>
    ...
  <option value="WY">Wyoming</option>
</select>
```

Kuva 19. Vaihtoehtoja valinta-laatikkoon

Select2 luomaa valinta-laatikkoa on myös mahdollista muokata omien tarpeidensa mukaan. Se tarjoaa monipuolisen ja kattavan dokumentaation, mistä on helppo lähteä rakentamaan omanlaistaan valinta-laatikkoa.

## 3.6 Datatables

Datatables on ilmainen taulukkojen parantamiseen käytettävä työkalu, jossa on sisäänrakennettu ominaisuuksia, kuten järjestäminen, etsiminen ja numeroiminen. Datatables on jQueryn liittännäinen, joka hyödyntää useita hyviä jQueryn ominaisuuksia. Datatablesin käyttöönotto onnistuu lisäämällä head-tagien sisään linkitykset. Itse datatablesin taulukko rakennetaan table-tagien sisään käyttämällä thead- ja tbody-tageja. (Datatables, 2021.) Tämän jälkeen täytyy vielä alustaa taulukko JavaScriptillä (Kuva 20).

```
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.23/css/jquery.dataTables.css">
<script src="https://code.jquery.com/jquery-3.5.1.js"></script>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script type="text/javascript" charset="utf8"
 src="https://cdn.datatables.net/1.10.23/js/jquery.dataTables.js">
</script>
<body>
    <table id="table_id" class="display">
        <thead>
            <tr>
                <th>Column 1</th>
                <th>Column 2</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Row 1 Data 1</td>
                <td>Row 1 Data 2</td>
            </tr>
            <tr>
                <td>Row 2 Data 1</td>
                <td>Row 2 Data 2</td>
            </tr>
        </tbody>
    </table>
</body>
<script>
$(document).ready( function () {
    $('#table_id').DataTable();
} );
</script>
</html>
```

Show 10 entries Search:

| Column 1 | Column 2 |
|---|---|
| Row 1 Data 1 | Row 1 Data 2 |
| Row 2 Data 1 | Row 2 Data 2 |

Showing 1 to 2 of 2 entries Previous 1 Next

Kuva 20. Datatables esimerkkitaulukko

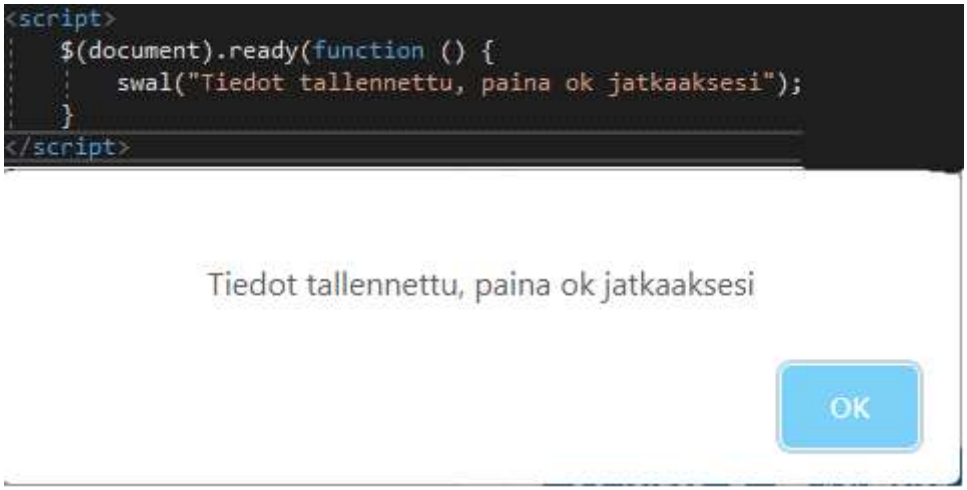
Datatables sisältää HTML-taulukoiden hallintaa helpottavia toimintoja, esimerkiksi suodatuksen ja näytettävien rivien määrän. Suodatuksella voidaan hakea tietyillä hakusanoilla suoraan taulukosta haluttua tietoa. Näytettävien rivien määrällä voidaan valita halutaanko enemmän vai vähemmän näkyviä rivejä samanaikaisesti. Nämä helpottavat tietojen hakemista nopeasti taulukosta.

## 3.7 SweetAlert

SweetAlert on työkalu ponnahdusikkunoiden nykyaikaisempaan ulosantiin. Se pitää sisällään JavaScript- ja CSS-tiedoston. Sitä voi käyttää  kolmella eri tyylillä. Ensinnäkin se on kirjasto, jota voi käyttää mihin tahansa www-projektiin, toiseksi se on suunniteltu

toimimaan Bootstrapin kanssa ja kolmanneksi sitä voidaan käyttää myös Android-projekteissa.

SweetAlertin käyttöönotto tapahtuu linkittämällä se head-tagien sisään. SweetAlerttia voidaan tämän jälkeen kutsua JavaScriptissä, kunhan DOM on jälleen valmiina. Tämän jälkeen popup-alertista saadaankin jo paremman näköinen (kuva 21). (SweetAlert, 2021.)




Kuva 21. SweetAlert esimerkki

# 4 RAZOR PAGES

## 4.1 Yleistä

Razor pages on Microsoftin kehittämä tekniikka, jota käytetään ASP.NET ympäristössä ja se on uudempi ja yksinkertaistettu verkkosovellusten ohjelmointimalli. Se on vaihtoehtoinen ohjelmointimalli MVC-mallille (Model-View-Controller). Se poistaa suuren osan ASP.NET MVC:stä käyttämällä tiedostopohjaisen reitityksen lähestymistapaa. Se on julkaistu vuonna 2017 osana ASP.NET Core 2.0:aa. (Mike Brind, 2019.)

## 4.2 MVVM-malli

MVVM-malli tulee lyhenteestä Model-View-ViewModel (kuva 22). Sen avulla saadaan erotettua käyttöliittymä ohjelman muusta toimintalogiikasta. Sovellus jaetaan kolmeen eri luokkaan. View (näkymä) sisältää käyttöliittymän ja siihen liittyvän logiikan, ViewModel (näkymämalli) sisältää ohjelman tilan ja yhdistää näkymän ja mallin. ViewModel tarjoaa näkymälle tarvittavan rajapinnan toiminnoille ja tiedon esittämiselle. Model (malli) sisältää bisneslogiikan ja datan. ViewModel pääsee käsiksi Modeliin, mutta View ei pääse muuten kuin ViewModelin kautta. (Neha Sharma, 2019.)


Kuva 22. MVVM-toimintamalli (Neha Sharma, 2019)

### 4.2.1 Malli (model)

Malli edustaa todellista dataa ja/tai tietoja, joita käsitellään. Malli-esimerkkinä voidaan käyttää vaikka henkilökorttia, joka sisältää nimen, puhelinnumeron ja osoitteen.

Malli pitää sisällään tietoja, mutta ei käyttäytymistä tai palveluita, jotka manipuloivat informaatiota. Se ei ole vastuussa siitä, että tekstin muotoilu näyttää hyvältä näytöllä tai luettelon kohteiden hakemisesta etäpalvelimelta (listan jokainen kohta olisi todennäköisesti itse oma mallinsa). Bisneslogiikka on tyypillisesti pidetty erillään mallista ja koteloitu muihin luokkiin, jotka vaikuttavat malliin. Tämä ei aina pidä kuitenkaan paikkaansa, vaan esimerkiksi jotkut mallit voivat sisältää validointia.

Mallin pitäminen kokonaan ”puhtaana” on yleensä haaste. Tällä tarkoitetaan esimerkiksi sitä, että henkilökortissa voi olla viimeisin muokkauspäivä ja muokanneen henkilön tiedot. Tällä ei välttämättä ole mitään oikeaa tarkoitusta henkilökortille, mutta se on tarpeellinen tieto sille, kuinka mallia käytetään, jäljitetään ja jatketaan järjestelmässä. (Jeremy Likness, 2021.)

### 4.2.2 Näkymä (View)

Näkymä on ainut asia, minkä kanssa loppukäyttäjä on tekemisissä, koska se esittelee datan. Näkymässä voi muokata mallista annettua dataa, jotta se olisi visuaalisesti miellyttävämpi. Esimerkiksi päivämäärä voi olla tallennettu modeliin sekunnin tarkkuudella, mutta näkymässä siitä näkyy loppukäyttäjälle vain päivämäärä ilman kellonaikaa. Näkymässä voi olla käyttäytymistä, kuten käyttäjän syötteen hyväksyminen. Näkymä hallitsee syötteitä (näppäinpainalluksia, hiiren liikkeitä, kosketuseleitä yms.), joka lopulta manipuloi mallin ominaisuuksia.

MVVM-mallissa näkymä on aktiivinen, toisin kuin passiivinen näkymä, jolla ei ole tietoa mallista ja jota näkymämalli (viewmodel) manipuloi kokonaan. Näkymä MVVM-mallissa pitää sisällään käyttäytymistä, tapahtumia ja datan sidoksia, jotka viime kädessä edellyttävät tietoa taustalla olevasta mallista ja näkymästä. Vaikka nämä tapahtumat ja käyttäytymiset saatetaan yhdistää ominaisuuksiin, menetelmäkutsuihin ja komentoihin, näkymä on silti vastuussa omien tapahtumiensa käsittelystä eikä käännä tätä täysin näkymämallille. Näkymä ei ole vastuussa tilansa pitämisestä vaan se synkronoi tilansa näkymämallin kanssa.

### 4.2.3 Näkymämalli (ViewModel)

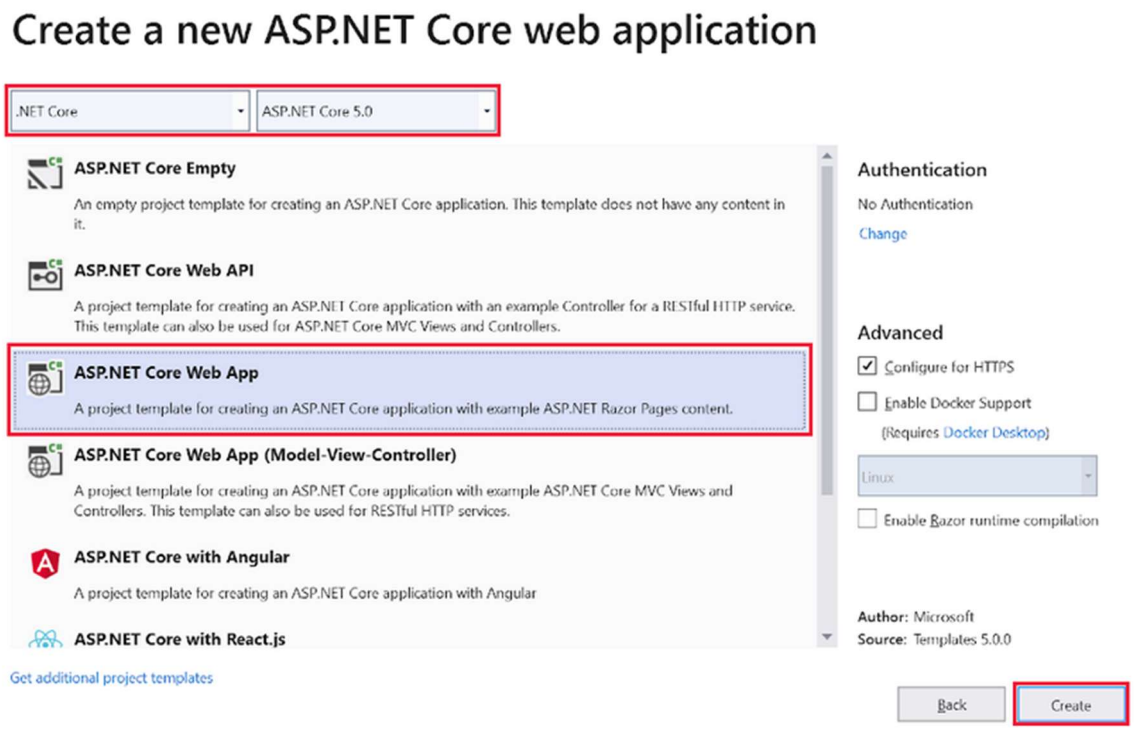
Näkymämalli on keskeisin osa tätä kolmikkoa, koska se pitää näkymän erillään mallista. Sen sijaan, että malliin saataisiin tieto käyttäjän näkymän päivämäärästä, jotta se muuntaisi päivämäärän näyttömuotoon, malli yksinkertaisesti pitää datan. Näkymä pitää alustettua päivämäärää ja näkymämalli toimii yhdistäjänä näiden kahden välillä. Näkymämalli saattaa ottaa syötteen näkymästä ja asettaa sen malliin tai se voi olla vuorovaikutuksessa palvelun kanssa mallin noutamiseksi, jonka jälkeen se kääntää ominaisuudet ja sijoittaa ne näkymään.

Näkymämalli paljastaa myös metodeja, komentoja ja muita kohtia, jotka auttavat ylläpitämään näkymän tilaa. Tämän lisäksi ne manipuloivat mallia näkymään kohdistuvien toimien seurauksena ja laukaisevat tapahtumia itse näkymässä.

## 4.3 Toimintaperiaate

Razor pages tekniikalla voidaan upottaa backendin logiikkaa cshtml tiedostoihin C#-kielellä. Sen sovelluskehys on monipuolinen. Se antaa kehittäjille hallinnan renderoidusta HTML:stä. Sitä käytetään kahdensuuntaiseen datan sidontaan näkymän, kontrollerin ja mallin välillä. Se keskittyy käyttöliittymään sekä näkymien luontiin ja sen avulla pystytään työskentelemään Razor pagesin kanssa ilman, että on MVC:stä tietoja ja taitoja.

Razor pagesin käyttöönotto tapahtuu Visual Studiossa luomalla uusi ASP.NET Core Web Application -projekti. Tämän jälkeen syötetään projektille nimi ja sijainti, minne tiedostot halutaan. Sitten valitaan ASP.NET Core ja versio, jota halutaan käyttää (kuva 23).




Kuva 23. ASP.NET Coren version valinta (Microsoft, 2021).

Pages-kansio pitää sisällään Razor pages-sivut ja tuetut tiedostot. Jokainen Razor pages-sivu muodostuu tiedostoparista, joka pitää sisällään .cshtml- ja .cshtml.cs-tiedostot. Cshtml-päätteinen tiedosto on HTML-kieltä, jota voi kehittää myös C#-kielellä Razor-syntaksia käyttäen. Cshtml.cs-päätteinen tiedosto pitää sisällään C#-kielistä ohjelmakoodia, joka käsittelee sivun tapahtumia.

Razor pagesin avulla C#-kieltä voidaan käyttää HTML-sivulla. Käyttö aloitetaan @-merkillä, jonka jälkeen koodi kirjoitetaan {}-sulkujen sisään, eli @{ ... }. Muuttujiin voidaan viitata suoraan sivulla @-merkillä. Razor pages mahdollistaa kutsujen käytön suoraan sivulla MVVM:ää hyödyntäen ilman, että se käy viewmodelin kautta (Kuva 24).

```
@{
    var hello = "Tervetuloa!";
}

<h1>@hello</h1>
<h2>Kellonaika serverillä on @DateTime.Now</h2>
```

Kuva 24. Kutsu suoraan HTML-sivulla

Suositeltu tapa kehittää Razor pages applikaatiota on pitää ohjelmakoodi palvelimella minimissä. Kaikki käyttäjän syötteiden tai tietojenkäsittelyyn liittyvät koodit tulisi sijoittaa PageModel-tiedostoihin. Tällä tavalla ohjelmakoodissa oleviin muuttujiin voidaan viitata suoraan HTML-sivulla. Ne jakavat myös saman tiedostonnimen, jonka erona on ainoastaan PageModel tiedostossa oleva .cs-pääte (kuva 25).

Esimerkki.cshtml

```
@page
@model ExampleModel
<div style="margin-top:30px;">
    <form method="post">
        <div>Name: <input asp-for="Name" /></div>
        <div><input type="submit" /></div>
    </form>
    @if (!string.IsNullOrEmpty(Model.Name))
    {
        <p>Hello @Model.Name!</p>
    }
</div>
```

Esimerkki.cshtml.cs

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
namespace RazorPages.Pages
{
    public class ExampleModel : PageModel
    {
        [BindProperty]
        public string Name { get; set; }
    }
}
```

Kuva 25. Luokka ja HTML

Page-kutsun lisäksi täytyy siis kutsua myös modelia, jolla määritellään oikea malli sivulle: tässä tapauksessa ExampleModel. Kuvassa 25 nähdään miten Name-muuttujaa kutsutaan HTML-sivulla.

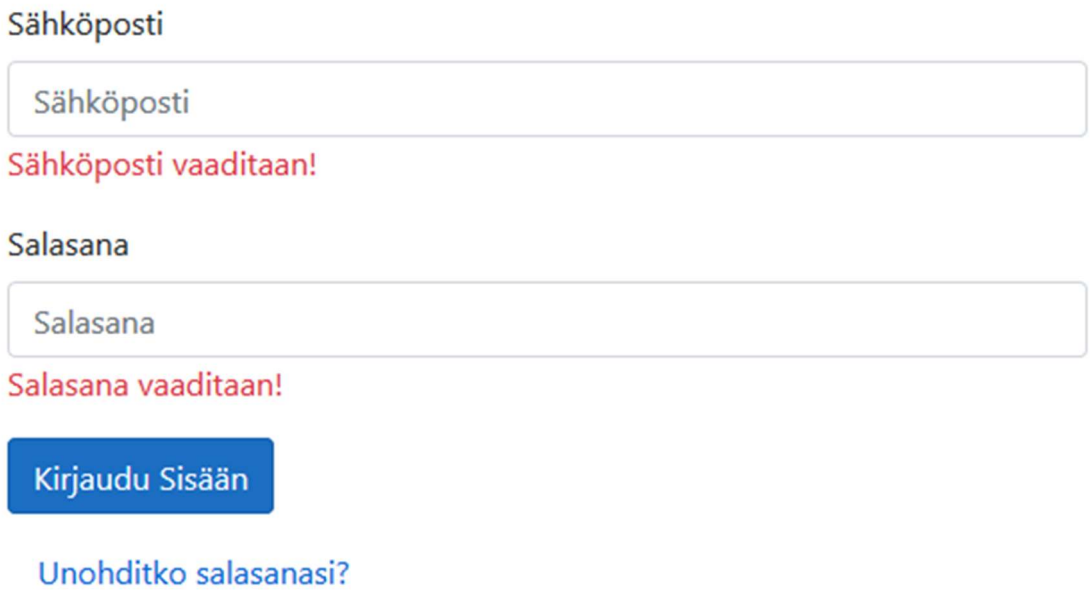
# 5 TOIMEKSIANTO

## 5.1 Yleistä

Oppimisportaali on rakennettu Visual Studiossa käyttämällä Razor pagesia. Tietokannat on rakennettu Visual Studion SQL Server Exploreria käyttämällä. Näiden lisäksi on käytetty tarvittavia CSS- ja JavaScript-kirjastoja.

## 5.2 Kirjautuminen

Kirjautumissivulla on tervetulotoivotus sivulle, sekä ylhäällä navigointipalkissa ”Kirjaudu sisään” painike ja kielen valinta valikko. Kielen valinnassa on vaihtoehtoina Suomi ja Englanti. Kirjaudu sisään-painikkeesta kirjaudutaan sisään. Sisäänkirjautumisessa käytetään tunnuksena sähköpostiosoitetta ja salasanaa. Sisäänkirjautuessa tiedot tarkistetaan tietokannasta ja kun tiedot ovat oikeat, päästään kirjautumaan sisään. Tietojen ollessa väärät, ilmoittaa sovellus joko virheellisestä sähköposti-osoitteesta tai salasanasta. Mikäli salasana on unohtunut, on tarjolla vielä ”Salasana unohtunut?”-linkki, mitä kautta pääsee syöttämään sähköpostinsa, johon salasana sitten lähetetään (kuva 26).



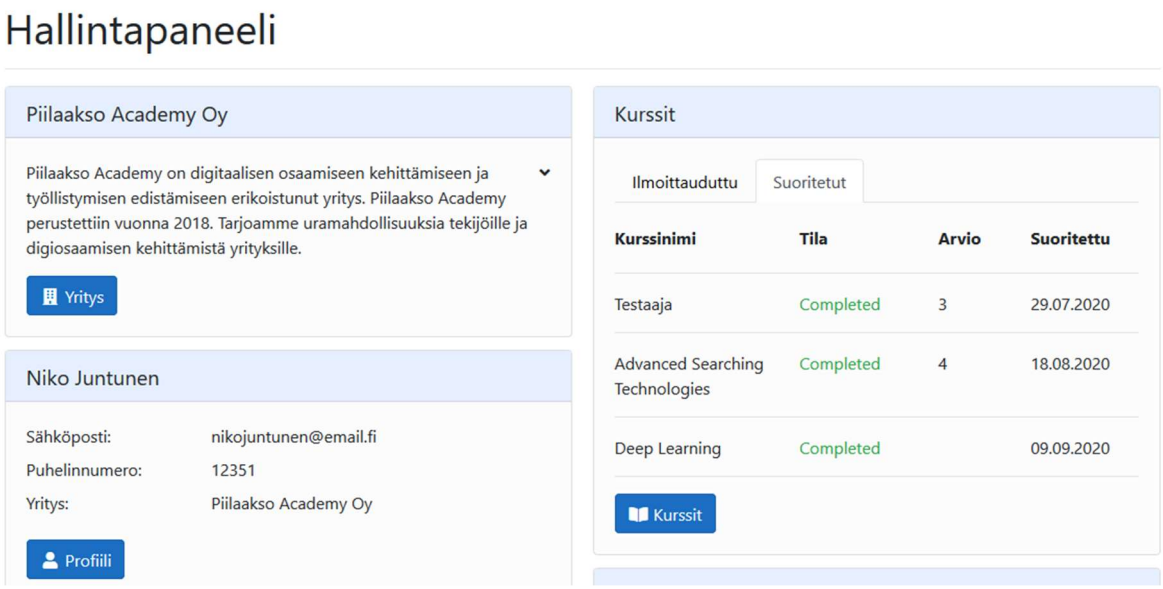

Kuva 26. Kirjautuminen

Käyttäjän rekisteröinti tapahtuu sivun ylläpitäjän toimesta (superadmin) tai työnantajan puolesta (admin). Käyttäjän tietoihin vaaditaan sähköposti, etu- ja sukunimi, puhelinnumero, yritys sekä salasana ja sen vahvistus. Salasanan käyttäjä pääsee vaihtamaan ensikirjautumisensa jälkeen.

## 5.3 Hallintapaneeli

Hallintapaneeli (dashboard) toimii käyttäjän etusivuna sisäänkirjautumisen jälkeen. Täältä käyttäjä löytää nopeasti tärkeimmät valikot, tulevat tapahtumat ja näkee tärkeimpiä tietoja omasta kehityksestään.

Hallintapaneelissa on käytetty Bootstrapin kortteja (card), jotta se olisi selkeämpi ja jokaisella osiolla olisi oma paikka. Jokainen kortti toimii myös ”pikavalikkona”, eli jokaisesta kortista pääsee nopeasti sen osion sivulle johon haluaa. Ylläpitäjille ja työnantajille on erikseen myös pikavalikko-kortti, jota kautta pääsee nopeasti haluamalleen sivulle (kuva 27).




Kuva 27. Hallintapaneeli

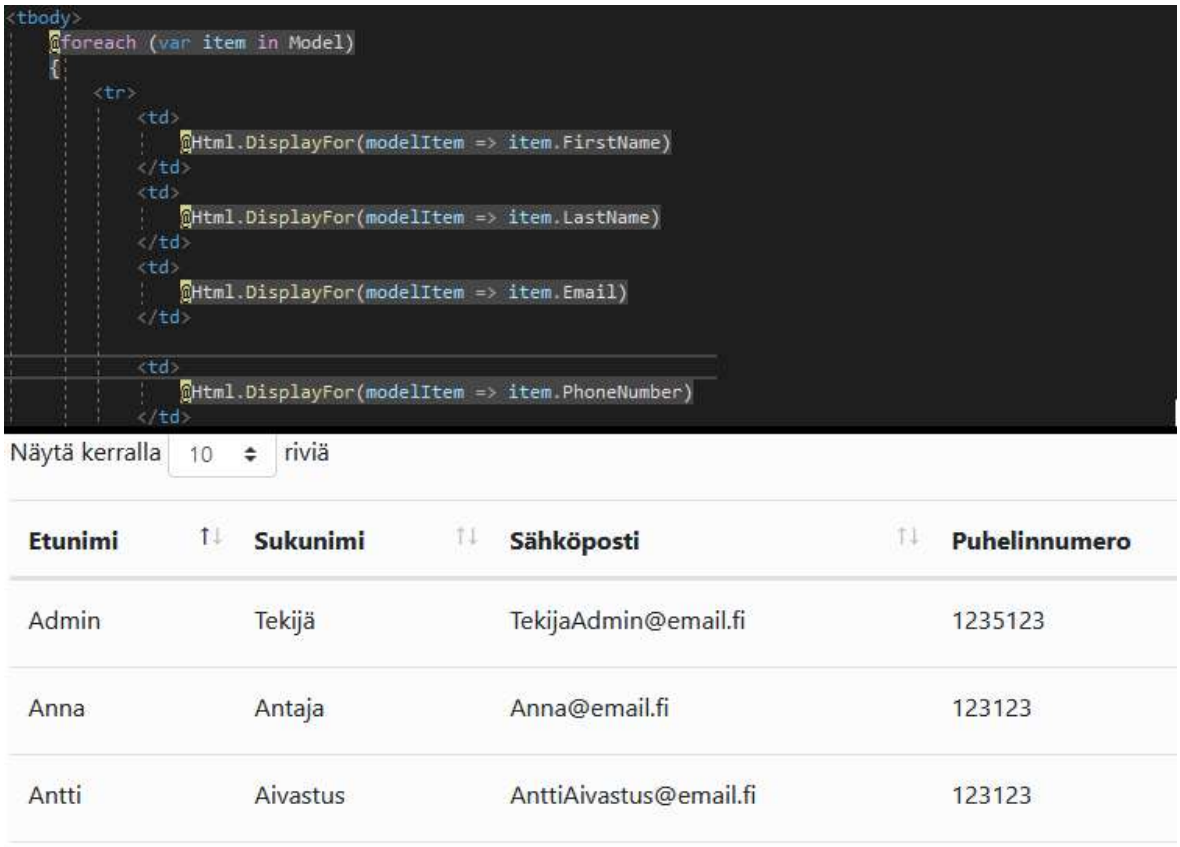
Hallintapaneelin pohjaratkaisu on jaettu kahteen, jotta sivulle mahtuisi enemmän tietoa kerralla ruudulle. Kortit menevät allekkain näytön koon pienentyessä tarpeeksi pieneksi responsiivisuuden ansiosta, jotta helppolukuisuus säilyisi.

## 5.4 Työntekijät

Työntekijät-sivulla on lista työntekijöistä. Ylläpitäjä näkee kaikkien yrityksien työntekijät, kun taas työnantaja näkee työntekijöistä vain ne, jotka kuuluvat samaan yritykseen, joka on työnantajalla aktiiviseksi valittuna. Ylläpitäjä ja työnantaja näkevät näiden lisäksi painikkeen, josta pääsee luomaan uusia käyttäjiä. Sen lisäksi he näkevät erillisen ”Tiedot”-painikkeen työntekijän tietojen oikealla puolella, josta pääsee tarkastelemaan työntekijän tietoja. Ylläpitäjä näkee kaikkien yrityksien työntekijät ja heidän tiedot, kun taas työnantaja näkee vain oman yrityksensä työntekijöiden tiedot. Työnantajan kuuluessa

useampaan yritykseen samanaikaisesti, on hänelle erikseen valikko, josta käyttäjä valitsee ”aktiivisen” yrityksen ja työntekijät näytetään silloin aktiivisesta yrityksestä. Työnantaja voi vaihdella aktiivista yritystä omien yritystensä välillä koska tahansa.

Henkilötietojen kutsut toteutetaan HTML-puolella, joka keskustelee modelin kanssa. HTML:ssä kutsutaan modelia, joka sitten hakee tarvittavat tiedot tietokannasta ja tulostaa ne sen jälkeen HTML-sivulle (kuva 28).




Kuva 28. Henkilötietojen haku ja tulostus

Jokaisella sivulla, jolla näytetään lista tiedoista, käytetään Datatablesin listausta. Datatablesin avulla saadaan helposti luotua helppolukuinen lista, josta on mahdollista myös hakea eri tietoja, sekä erikseen voidaan määritellä montako riviä kerrallaan näytetään. Käyttäjä voi myös valita, minkä rivin mukaan mennään nousevassa tai laskevassa järjestyksessä.

## 5.5 Tarkka haku

Tarkkaa hakua käytetään silloin, kun pitää hakua kohdentaa useammalla eri hakukriteerillä, esimerkiksi haetaan työntekijää, jolla on taidot tietyllä tasolla ja hän kuuluu tiettyyn ryhmään. Tarkka haku on suunnattu käytettäväksi vain ylläpitäjälle ja

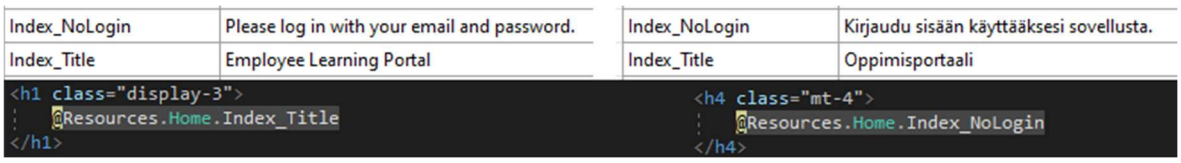
työnantajalle. Ylläpitäjä voi erikseen etsiä mihin vain yritykseen kuuluvia henkilöitä ja hän näkee näiden kaikkien tiedot. Työnantaja sen sijaan näkee vain oman aktiivisen yrityksensä henkilöiden tiedot ja muihin yrityksiin kuuluvat näyttävät vain tiettyjä tietoja. Taitoja voidaan hakea yksittäisenä tai useampaa kerralla ja näihin hakuihin voidaan määritellä, miltä väliltä taitotasoa halutaan etsiä arvosanoilla 0-5.

## 5.6 Profiili

Profiilin kautta pystyy käyttäjä tarkastelemaan omia tietojaan ja taitojaan. Tätä kautta näkyvät kaikki omat käyttäjätiedot, oma yritys/yritykset, ryhmät joihin käyttäjä kuuluu, käyttäjän taidot, sekä käyttäjän sertifikaatit. Tietojaan pääsee myös muokkaamaan tätä kautta. Niin kuin muillakin sivuilla, myös tällä sivulla on hyödynnetty Bootstrapin kortteja, jotta on helposti saatu sivua selkeämmäksi ja jokainen profiili-sivulle kuuluva osio jaettua omiin kortteihinsa.

## 5.7 Käännökset

Oppimisportaali rakennettiin kahdelle kielelle: Suomi ja Englanti, joten siinä hyödynnetty XML:n resurssi- (resources) tiedostoja. Resurssit-osio löytyy Visual Studion solution explorer-ikkunasta alimmaisena projektista. Tänne tulee .resx-tiedostot, joihin luodaan käännökset. Näitä käännöksiä kutsutaan HTML-sivulla (kuva 29). Kuvassa nähdään vasemmassa reunassa englanninkielisiä käännöksiä, sekä oikealla suomenkielisiä käännöksiä. Kuvan alhaalla nähdään, että molemmissa on samat käännös-kutsut kielestä riippumatta. Kun sivulla on valittu kieli, osaa ohjelma valita oikean käännöksen näytettäväksi. Oletuskielenä on Englanti.




Kuva 29. Käännösten määrittely

Käännökset vaativat toimiakseen lisäksi määrityksiä, jotta saadaan käännökset vaihtumaan, kun yläpalkista valitaan haluttu kieli. Nämä määritykset lisätään startup.cs-tiedostoon lisäämällä lokaalisaatio-määrityksiä (kuva 30). Kuvassa 30 lisätään ensin polku, mistä käännökset haetaan. Tämän jälkeen lisätään muuttuja, josta tehdään lista, jotta saadaan kaksi kieltä liitettyä siihen, joiden välillä kieliä vaihdetaan. Oletuskieleksi asetetaan englanti.

```
services.AddLocalization(options => options.ResourcesPath = "Resources");
services.Configure<RequestLocalizationOptions>(
options =>
{
    var supportedCultures = new List<CultureInfo>
    {
   new CultureInfo("en-GB"),
   new CultureInfo("fi-FI"),
    };
    options.DefaultRequestCulture = new RequestCulture(culture: "en-GB", uiCulture: "en-GB");
    options.SupportedCultures = supportedCultures;
    options.SupportedUICultures = supportedCultures;
    services.AddSingleton(options);
});
```

Kuva 30. Määritykset startup.cs-tiedostossa

## 5.8 Ylläpito

Yläpalkista löytyy oma valikko superadmineille ja admineille. Tämän valikon nimi on ylläpito ja se on rakennettu dropdown-valikoksi. Ylläpitovalikosta pääsee nopeasti ja tehokkaasti eri sivuille, joihin tarvitsee tehdä muutoksia, lisäyksiä tai poistoja. Valikon ideana on nopeuttaa ylläpitäjien työskentelyä. Ylläpidolle olevan valikon näkyvyys tarkistetaan koodissa ensin katsomalla onko käyttäjä kirjautunut sisään ja sen jälkeen tarkistamalla käyttäjän rooli (kuva 31).

```
@if (SignInManager.IsSignedIn(User))
{
    @if (User.IsInRole("Admin") || User.IsInRole("Superadmin"))
```

Kuva 31. Roolin tarkistus

Käyttäjän sisäänkirjautumisen ja roolin tarkistamisen jälkeen, tulee navigointipalkkiin näkyviin roolin mukaiset valikot. Ylläpitäjille valikoita näkyy enemmän.

### 5.8.1 Yritykset

Ylläpidon dropdown-valikon kautta löytyy yritykset-sivu. Yritykset-sivun tarkoituksena on pitää ylläpitäjille listaa yrityksistä, sekä mahdollistaa niiden tietojen katsomisen tätä kautta. Superadmin voi tätä kautta lisätä lisää yrityksiä. Yrityksen tietoihin kuuluu esittelyteksti yrityksestä, lista yrityksen työntekijöistä (näkyy vain superadminille), sekä yrityksen määrittelemät taitotavoitteet. Yrityksien ideana on jakaa käyttäjiä omiin yrityksiinsä, jotta pystytään helposti pitämään kirjaa siitä, mihin yritykseen kukin käyttäjä kuuluu.

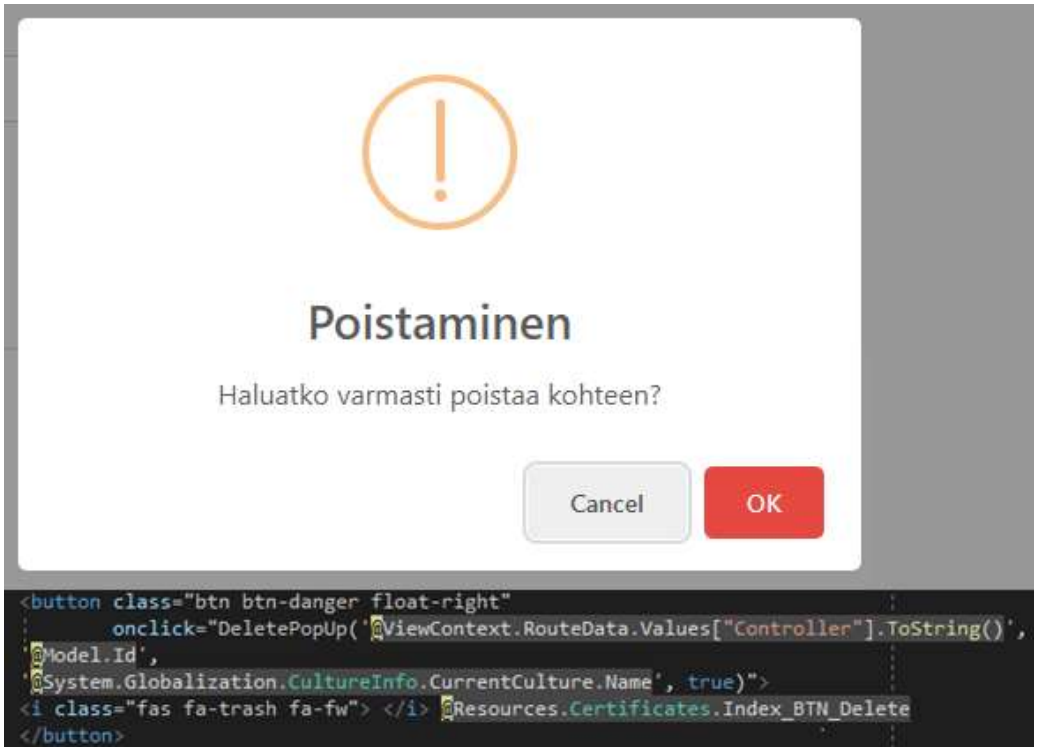
Käyttäjä voi kuulua useampaan yritykseen samanaikaisesti. Käyttäjä voi olla toisessa yrityksessä admin ja toisessa normaali käyttäjä. Tämän vuoksi käyttäjä ei voi nähdä siinä yrityksessä olevia työntekijöitä, joissa hän on itse sovelluksessa normaali käyttäjä.

Navigointi-palkkiin on lisätty tätä varten erikseen dropdown-valikko, jonka kautta käyttäjä voi valita itselleen ”aktiivisen”-yrityksen. Tämän valikon kautta sovellus vertailee, mitkä oikeudet käyttäjällä on valitussa yrityksessä ja sen mukaan käyttäjä pääsee sovellusta käyttämään.

### 5.8.2 Sertifikaatit

Sertifikaatti on dokumentti, joka todistaa tietyn taidon löytymisen tietyltä henkilöltä. Sertifikaatit toimivat ylläpidon dropdown-valikon kautta. Sertifikaatit-sivulla näkyy lista sertifikaateista. Superadminit ja adminit voivat lisätä uusia sertifikaatteja, jos tiettyä sertifikaattia ei sovelluksesta jo löydy. He voivat myös myöntää sertifikaatteja käyttäjille tätä kautta. Sertifikaateille asetetaan myöntämispäivä ja mikäli sertifikaatissa on vanhenemispäivä, syötetään myös sekin.

Sertifikaattien muokkaamissivun kautta sertifikaattia voidaan vielä muokata, mikäli sen luonnissa on tapahtunut virhe tai muu vastaava. Tätä kautta sertifikaatti voidaan myös poistaa tarvittaessa. Poista-nappia painamalla ilmestyy SweetAlertin popup-ikkuna, joka varmistaa halutaanko kyseinen sertifikaatti varmasti poistaa (kuva 32).



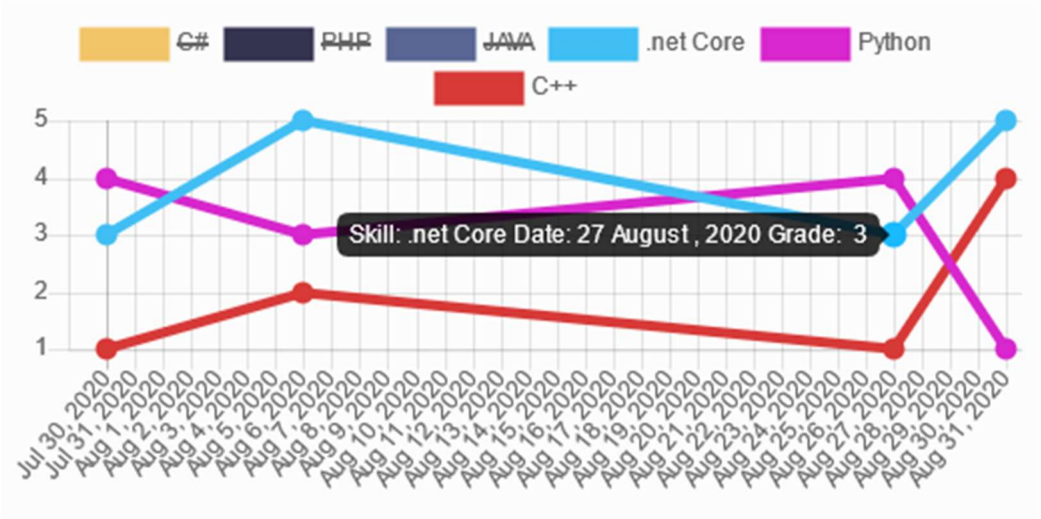

Kuva 32. SweetAlert popup

Kuvasta nähdään myös ponnahdusikkunan ohjelmointia. Onclick-tapahtumasta lähtee tieto, jonka avulla näkymä lukee, mistä controllerista tieto pitää poistaa. Se lukee tiedoston id:n ja vertaa sitä tietokannassa olevan rivin id:n kanssa ja poistaa sen, kun OK-

nappia klikataan. Alin määrittely määrää sen, mitä kieltä SweetAlert popup-ikkunassa näytetään.

## 5.9 Taitokehitys

Taitokehitystä seurataan omalla sivullaan viivakaavion avulla. Täältä käyttäjä näkee miten kaikki taidot ovat kehittyneet. Käyttäjä voi valita miltä kuukaudelta ja vuodelta haluaa taitoja tarkastella, sekä hän voi myös erikseen valita, mitä taitoja kaaviossa näytetään. Viivakaavion viivat osuvat palloihin, jotka ovat arviointipäiviä. Viemällä hiiren osoittimen pallon päälle, käyttäjä näkee taidon, tarkan arviointipäivän, sekä arvioinnin numeron (kuva 33).

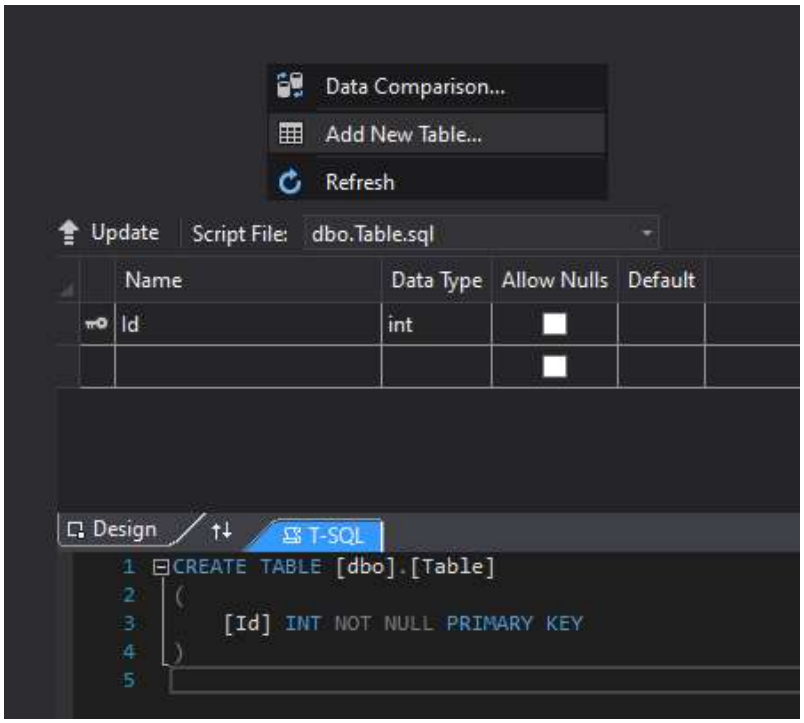



Kuva 33. Viivakaavio

Taitokehitystä voi seurata myös käyttäjätaidot sivulla, josta käyttäjä näkee omat taitonsa, niiden tämän hetkisen tason ja tavoitteen. Sivulla näkee myös parhaan taitonsa, huonoimman taitonsa, sekä taitojensa keskiarvon. Superadmin ja admin voivat käydä tätä kautta antamassa arvion käyttäjän taidoista.

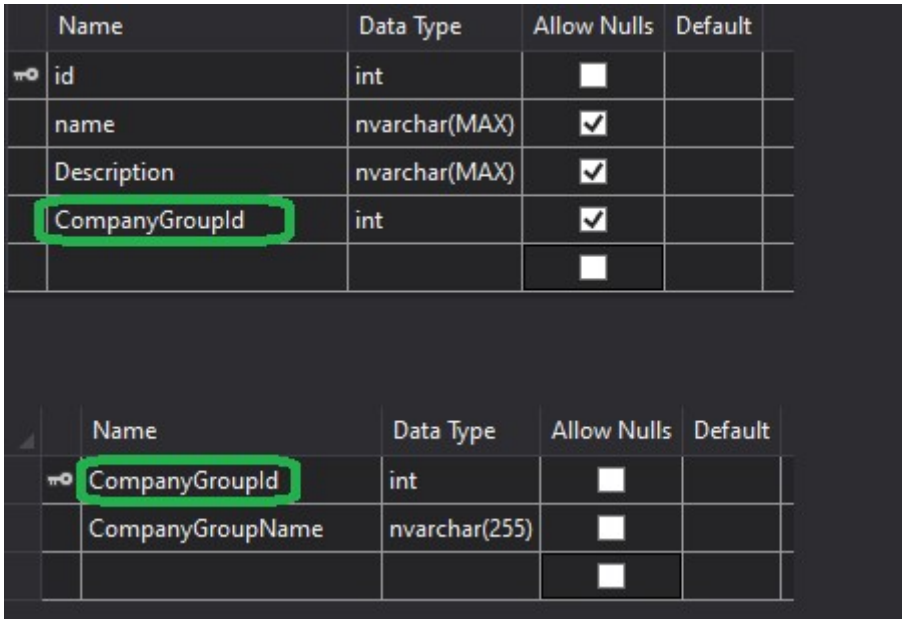
## 5.10 Tietokannat

Tietokantojen rakentaminen toteutettiin Visual Studion SQL Server Object Explorerin kautta. Tätä kautta on luotu taulut käyttämällä new table-komentoa, joka luo valmiiksi pohjan id-avaimen kanssa (kuva 34).



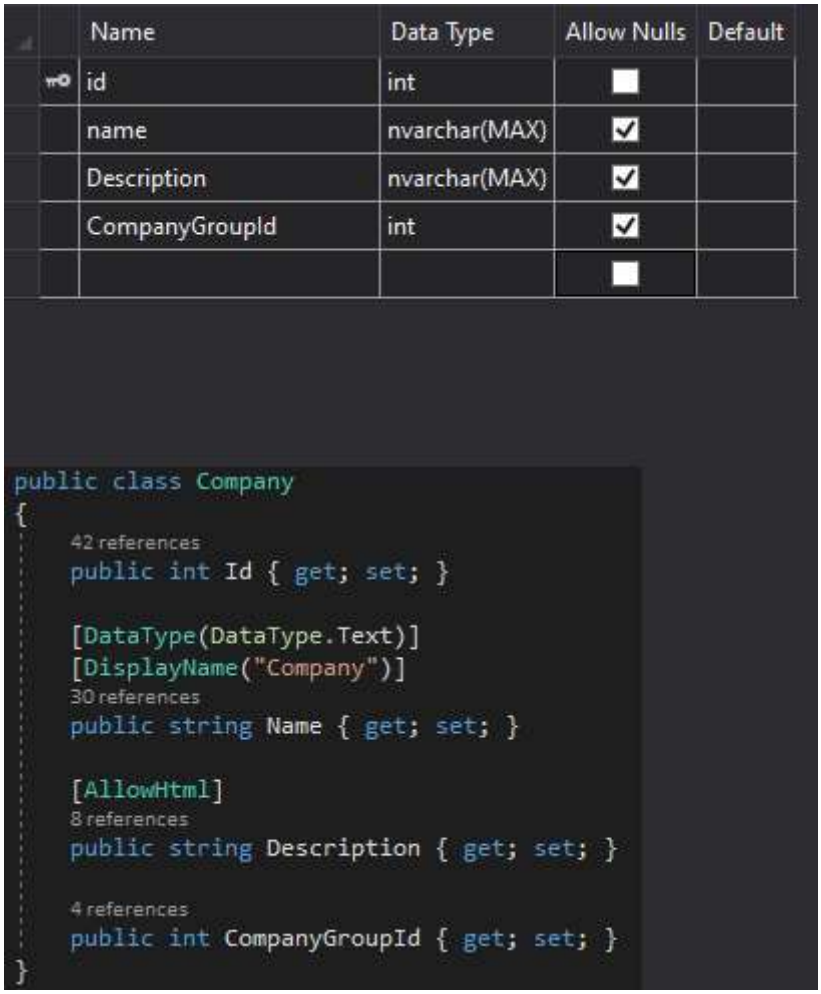

Kuva 34. Taulun lisäys tietokantoihin

Kuvassa nähtävään pohjaan lisättiin tarvittavat avaimet, joita mallissa (model) tarvitaan. Näillä avaimilla saadaan tallennettua, haettua, muokattua ja poistettua tarvittavat tiedot tietokannoista. Linkitys eri tietokantojen tietojen kesken on rakennettu myös tietokantojen kautta. Esimerkiksi seuraavassa kuvassa nähdään kuinka molemmissa tauluissa on avain ”CompanyGroupId”, jonka mukaan sovellus osaa lisätä tarvittaessa tietoja molempiin tietokantoihin (kuva 35).




Kuva 35. CompanyGroupId-linkitys

Mallissa (model) täytyy olla jokainen avain määritelty erikseen. Avaimet joissa ei hyväksytä tyhjiä (null), täytyy arvot antaa jossai vaiheessa, ennen kuin tieto tallennetaan tietokantaan. Esimerkiksi seuraavassa kuvassa nähdään, miten Company-taulu ja Company-malli on rakennettu (kuva 36).



| | Name | Data Type | Allow Nulls | Default |
|---|---|---|---|---|
| 🔑 | id | int | ☐ | |
| | name | nvarchar(MAX) | ☑ | |
| | Description | nvarchar(MAX) | ☑ | |
| | CompanyGroupId | int | ☑ | |
| | | | ☐ | |

```
public class Company
{
    42 references
    public int Id { get; set; }

    [DataType(DataType.Text)]
    [DisplayName("Company")]
    30 references
    public string Name { get; set; }

    [AllowHtml]
    8 references
    public string Description { get; set; }

    4 references
    public int CompanyGroupId { get; set; }
}
```

Kuva 36. Company-taulu ja -malli

Kuten kuvasta nähdään, taulun avaimet täsmäävät mallin muuttujiin. Erikseen taulun avaimiin voidaan määritellä, mitä tyyppiä data oletettavasti tulee olemaan. Esimerkiksi id on int eli numeerinen arvo ja name voi sisältää mitä vain Unicode-merkkejä.

# 6 YHTEENVETO

Työn tavoitteena oli saada luotua oppimisportaalin pilottiversio Razor pages-ohjelmistokehystä käyttämällä. Siitä täytyi rakentaa mahdollisimman toimiva ratkaisu työnantajalle ja työntekijälle. Lisätavoitteena oli hyötykäyttää CSS- ja JavaScript-kirjastoja, jotta sivustosta saadaan moderni.

Oppimisportaalin pilottiversiossa saatiin lähes kaikki ominaisuudet toimiviksi, mutta lisättäviä ominaisuuksia myös vielä jäi. Tämän vuoksi nykyisiäkin ominaisuuksia todennäköisesti joutuu vielä tulevaisuudessa muuttamaan. Tällä hetkellä kaikki projektin frontendin puolella olevat haut tietokannasta backendin kautta toimivat. Rooleissa ainoastaan yritysryhmäadmin-roolin toimivuudessa jäi parantamisen varaa.

Oppimisportaaliin jäi vielä muutama ongelma, joiden ratkaiseminen on seuraava askel. Yritysryhmäadmin-roolin hiominen täytyy toteuttaa useamassa paikassa kuntoon. Käyttäjät voivat tällä hetkellä tarkastella toisten tietoja lisäämällä toisen henkilön id-arvon URL-osoitteeseen. Tämä on suuri tietoturvariski, minkä vuoksi sovellus ei ole vielä julkaisukelpoinen.

Oppimisportaalin kehitys jatkuu vielä. Käyttökokemuksien perusteella on helppo lähteä tekemään muutoksia, jotta sovelluksesta saataisiin vieläkin toimivampi ratkaisu. Oppimisportaalia täytyy kehittää lisää, sekä puutteet täytyy korjata. Oppimisportaalin kehittämistä jatkaa tällä hetkellä toinen tiimi.

# LÄHTEET

Bootstrap. Build fast, responsive sites with Bootstrap. Viitattu [17.3.2021]. Saatavissa: https://getbootstrap.com/

Chartjs. Chart.js. Viitattu [17.3.2021]. Saatavissa: https://www.chartjs.org/

Datatables. Add advanced interaction controles to your HTML tables the free & easy way. Viitattu [17.3.2021]. Saatavissa: https://datatables.net/

Font Awesome. Font Awesome. Viitattu [17.3.2021]. Saatavissa: https://fontawesome.com/

Piilaakso Academy. Piilaakso Academyn tarina. Viitattu [17.03.2021]. Saatavissa: https://www.piilaakso.com/yritys/

Jeremy Likness, 2014. Model-View-ViewModel (MVVM) Explained. Viitattu [17.03.2021]. Saatavissa: https://www.wintellect.com/model-view-viewmodel-mvvm-explained/

Jquery. Write less, do more. Viitattu [17.3.2021]. Saatavissa: https://jquery.com/

Microsoft, 2021. Introduction to Razor Pages in ASP.NET Core. Viitattu [17.3.2021]. Saatavissa: https://docs.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-5.0&tabs=visual-studio

Mike Brind, 2019. Razor Pages. Viitattu [17.3.2021]. Saatavissa: https://www.learnrazorpages.com/razor-pages

Neha Sharma, 2019. MVVM Design Pattern for building robust and scalable iOS apps. Viitattu [17.3.2021]. Saatavissa: https://medium.com/codewave/mvvm-design-pattern-c5d9f4a10758

Select2. The jQuery replacement for select boxes. Viitattu [17.3.2021]. Saatavissa: https://select2.org/

SweetAlert. A beautiful replacement for messages. Viitattu [17.3.2021]. Saatavissa: https://sweetalert.js.org/