



YourTone Startup

Tradenomi, tietojenkäsittelyn tutkinto

Hautala Niko

Haaga-Helia ammattikorkeakoulu

2021

Amk-opinnäytetyö

Tiivistelmä

Tekijä(t)

Niko Hautala

Tutkinto

Tradenomi, Tietojenkäsittely

Raportin/Opinnäytetyön nimi

YourTone Startup

Sivu- ja liitesivumäärä

21 + 3

Tässä opinnäytetyössä käsittelemme Django kehukseen rakennettua verkkopalvelua. Opinnäytteen tavoitteeksi valittiin miten luoda käyttäjäystävällinen polku rekisteröinnistä profiiliin luomiseen. Rajauksen ulkopuolelle jäi näin ollen startup yrityksen kehitys ja muu verkkopalvelun luominen.

YourTone on verkkopalvelu, joka toimii hakukoneen tavoin. Sivustolta voi etsiä musiikin yksityisopetusta ja palvelun tavoite on lisätä musiikkia opettavien näkyvyyttä ja työllistymistä sille tarkoitetun hakupalvelun avulla. Liikeidean kannattavuutta on mitattu markkinatutkimuksella, jonka avulla on selvitetty palvelun tarvetta sekä minkälaisia ominaisuuksia sen tulisi sisältää.

Verkkopalvelun luomisessa on pääasiallisesti käytetty Django kehystä ja sitä tukevia työkaluja sekä teknologioita kuten Bootstrap-kirjasto. Verkkosovellus on luotu bootstrapping-ideologialla, joten siinä käytetyt työkalut ja teknologiat on valittu tukemaan rajattuja resursseja. Työskentelyä tukevia työkaluja on esimerkiksi riippuvuuksienhallintaan Docker, sivuston toimivuuden raportointiin Sentry ja suunnitteluun Figma.

Verkkosovelluksen arkkitehtuuri on alustettu Django projektilla, joka toimii eritoten modulaarisesti. Sovelluksen koodi järjestetään eri moduuleihin niiden käyttötarkoitusten mukaan. Kun kyseessä on palveluun rekisteröinti, on tietoturvallisuus otettu huomioon. Django all auth kirjasto antaa perusteet toteutukseen.

Asiasanat

verkkopalvelu, ohjelmointi, startup-yritykset

Sisällysluettelo

1	<i>Johdanto</i>	1
2	<i>Projektin lähtökohdat</i>	2
2.1	Tiimi	3
2.2	Tarve	3
2.3	Tavoite ja rajaus	4
2.4	Työkalut	4
2.4.1	Git ja GitLab	4
2.4.2	Visual Studio Code (VSC)	5
2.4.3	Figma	6
2.4.4	Poedit	6
2.4.5	Sentry	6
2.4.6	Google Drive	6
2.4.7	Trello.....	7
2.4.8	Kommunikaatiovälineet	7
2.5	Teknologiat	7
2.5.1	Django	7
2.5.2	Bootstrap.....	9
2.5.3	Python	9
2.5.4	Riippuvuuksien hallinta ja Docker	9
3	<i>Sovelluksen yleiskuva ja rakenne</i>	11
3.1	Rakenne ja arkkitehtuuri	11
3.2	Sovelluksen osien koodaus	11
3.3	Sovelluksen tietoturvallisuus	11
4	<i>Kehittämistyö</i>	12
4.1	Projektin alku	12
4.2	Projektiin tehtyjä töitä	12
4.3	Vaatimusmäärittely	16
4.4	Responsiivisuus	17
4.5	Monikielisyys	17

5	<i>Pohdinta</i>	18
5.1	Palvelu.....	18
5.2	Asenne.....	18
5.3	Oppi.....	19
5.4	Tavoitteet	19

1 Johdanto

Musiikin yksityisopetuksen etsiminen ja soittotuntien markkinointi voi olla vaikeaa. Opetusta on hankala löytää Google-haulla, eikä tarjonnan vertailu ole helppoa.

Ratkaisuna tähän ongelmaan aloimme rakentamaan YourTone-hakupalvelua.

YourTone on palvelu, jossa musiikin yksityisopettajat voivat markkinoida itseään selkeän opettajan profiilin avulla. Opettajaa etsivät hakevat sivustolta opettajaa paikkakunnan ja soittimen perusteella, jolloin he löytävät kaikki hakua vastaavat, palvelussa aktiiviset opettajat.

Valitsin opinnäytetyöni aiheeksi YourTone-hakukoneen kehittämisen; tarkemmin rajattuna sen, miten luoda käyttäjäystävällinen polku rekisteröinnistä profiilin luomiseen.

Työssä opettelen Django ohjelmointi kehystä ja sen käyttöä, sekä käytännössä Fullstack koodausta. Muita käytettyjä ohjelmia ovat Docker, GitLab, sekä Visual Studio Code. Tavoitteena oli luoda vuoden 2021 loppuun mennessä MVP (minimum viable product), jotta hakukone saadaan markkinoille.

YourTone-palvelulla on kaksi asiakasryhmää: musiikin opetusta tarjoavat ja opetusta ostavat henkilöt. Toisen ryhmään kuuluvat potentiaaliset asiakkaat eli musiikin opiskelusta kiinnostuneet henkilöt. YourTone-palvelu tarjoaa näille ryhmille kohtamisalustan. Palvelun tarjoajina me huolehdimme, että tuotteen käyttäjillä on riittävästi näkyvyyttä.

Moderni ulkoasu on nykyaikana tärkeä, monet musiikinopetukseen kohdentuneet sivustot jäivät tältä osin hyvin alkeellisiksi.

Yrityksemme tarjoaman palvelun sivusto on toiminnassa ja löytyy osoitteesta yourtone.fi/.

2 Projektin lähtökohdat

Vuonna 2019 Bea Mattila etsi itselleen pianonsoiton opettajaa. Huomasimme silloin, että yksityisopettajan etsiminen ei ole helppoa. Henkilökohtainen tarpeemme johti siihen, että aloimme pohtimaan, miten voisimme keksiä ratkaisun tähän ongelmaan. Seuraavana askeleena teimme markkinatutkimuksen sekä aloitimme liiketoiminnan kehityksen.

Vuonna 2021 saimme tiimiimme kokeneen, Sérgio Isidoron. Jatkoimme idean kannattavuuden tutkimista kartoittamalla palvelun tarvetta musiikinopetusta tarjoavilta, sekä opetukselta kiinnostuneilta. Saimme selville, että monille opettajille markkinointi ja asiakkaiden tavoittaminen on ongelma, myös opetusta etsivät kokivat opettajien löytämisen ja vertailun haastavaksi.

Mattila loi pohjan liiketoimintasuunnitelmalle sekä teki selvityksen mahdollisista kilpailevista palveluista. Tämän jälkeen aloimme kehittää liiketoimintasuunnitelmaa pidemmälle kaikki kolme perustajajäsentä yhdessä.

Yhdessä myös määritimme projektillemme onnistumisen kriteerit;

- Hakukone optimointi, jotta asiakkaanamme olevat musiikinopettajat näkyvät, sekä meidän hakukoneessamme, että yleisten suurempien hakukoneiden hauissa.
- Markkinointi; palvelua tulee aktiivisesti markkinoida, jotta sekä sen että asiakkaiden näkyvyys saadaan taattua.
- Helppo rekisteröityminen; haluamme, että käyttäjälle on mahdollisimman pieni kynnyks aloittaa palvelun käyttö, sekä sen lopettaminen tarvittaessa, mallina toimivat suoratoisto palvelut.
- Maksutoiminta; käyttäjän tulee kyetä rekisteröityä palveluun ilman maksuja, sekä maksut ja muut toiminnot tulee olla ilmi ilman rekisteröitymistä
- Moderni ulkoasu; otimme mallia muutamista mielestämme toimivista sivustoista, jotka vastasivat mielikuvaamme helposta, selkeästä ja käyttäjäystävällisestä palvelusta.

Projektia alettiin ideoimaan ja kehittämään vuonna 2019 ja sivusto julkaistiin huhtikuussa 2022. Suurimman työn omaan osuuteeni tein vuoden 2021 lopulla ja vuonna 2022 olen keskittynyt sivuston kehittämiseen hieman pienemmällä panostuksella.

Projektilla oli monta neuvonantajaa, sen kehitystä oli tukemassa Haaga-Helian Startup valmentaja Teemu Ruohonen, Keuken uusyrityskeskus, sekä muutamia muita tahoja, joiden kanssa keskustelimme yritysideoista. Osallistuimme lisäksi, Ultrahackin, Hack the Music Industry, hackathoniin. Näin saimme kehittää yritysideoamme sekä sen esitystä hackathoniin osallistuneiden neuvonantajien kanssa.

Oma osuuteni on varsin monimuotoinen, 2019 idean synnyttyä, tein mm. markkinointikartoitusta tutkien, kuinka paljon muusikkoja ja musiikkipedagogeja valmistuu kouluista, hain lukuja myös bändirekisteristä. Olen ollut aktiivisessa roolissa idean kehityksessä ja toteutuksessa.

2.1 Tiimi

Tiimiimme kuuluvat seuraavat henkilöt:

Bea Mattila; päävastuualueina liiketoiminnan kehitys, visuaalisuus sekä markkinointi

Sérgio Isidoro; päävastuualueina sovelluksen kehitys sekä mentorointi

Niko Hautala; päävastuualueina sovelluksen kehitys sekä avustava toiminta

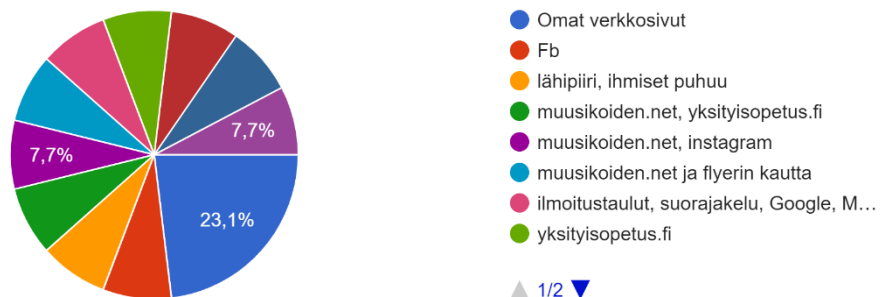
2.2 Tarve

Kartoitimme tarvetta kahden kyselyn avulla. Näiden kyselyiden avulla saimme tietää, että suunnittelemallemme palvelulle olisi käyttöä.

Musiikinopettajille suunnatusta kyselystä saimme 19 vastausta. Vastauksista kävi ilmi, että 44,4 % ei markkinoi yksityisiä musiikintuntejaan, 50 % kokee markkinoinnin haastavana, 58,8 % ei tavoita tarpeeksi asiakkaita ja 72,2 % olisi kiinnostunut tarkoitukseen luodusta hakukoneesta, lisäksi yksityisopettajien markkinointikanavat ovat hajallaan useassa eri lähteessä.

Miten markkinoit palveluitasi?

13 vastausta



Kuva 1, YourTone kysely musiikinopettajille ”Miten markkinoit palveluitasi?”

Kartoitimme myös mitä musiikin instrumenttia tai aihetta he opettavat tai haluaisivat opettaa, olisivatko he valmiita maksamaan tällaisesta musiikinopetuksen hakupalvelusta ja minkä verran, sekä ”Mitä haluaisit kertoa tai näyttää potentiaalisille oppilaille osoittaaksesi, että olet heille oikea valinta?”

Vastausten perusteella saimme selville, kuinka moni olisi valmis käyttämään tällaista palvelua ja mitä ominaisuuksia opettajan profiiliin olisi hyvä pitää sisällään, jotta se vakuuttaisi musiikin yksityisopettajat käyttämään palvelua.

Kyselyyn musiikin oppimisesta kiinnostuneille saimme 47 vastausta. Vastanneista 76,6 % kokee musiikin yksityisopetuksen haastavana löytää, 86,7 % etsii opetusta pääsääntöisesti googlen avulla, 85,1 % olisi valmiita ottamaan musiikin yksityisopetusta opiskelijalta tai harrastelijalta ja 93,6 % piti musiikinopetuksen hintavertailua tärkeänä. Näiden vastausten perusteella pystyimme selvittämään palvelun tarpeellisuutta ja tutkimaan sitä mitä palvelun tulisi pystyä tarjoamaan musiikin yksityisopetusta etsiville.

Muita kysymyksiä, joilla kartoitimme sivuston tarvetta, olivat muun muassa; mitä musiikin taitoa opettelet tai haluaisit opetella ja mitä ominaisuuksia otat huomioon ja haluat verrata, kun valitset musiikinopettajaa. Näiden kysymysten avulla pystyimme selventämään, minäläisiä musiikinopettajia tarvitsemme palveluun ja mitä opettajan profiiliin tulisi pitää sisälleen opetusta etsivän näkökulmasta.

2.3 Tavoite ja rajaus

Tavoitteena on luoda käyttäjä ystävällinen polku rekisteröinnistä profiiliin luomiseen.

Rajaukseen sisältyy;

- Rekisteröintilomake, jolla käyttäjä voi rekisteröityä palveluun.
- Rekisteröinnin jälkeinen sivu, jossa käyttäjä valitsee, haluaako opettaa musiikkia, vai opetella musiikkia.
- Käyttäjän profiiliin asetukset-sivu, josta näkee omat tiedot sekä oman opetusprofiilin tiedot ja pääsee muokkaamaan niitä.

2.4 Työkalut

Päädyimme käyttämään tämän sovelluksen luomiseen työkaluja, jotka tukevat projektimme tehokasta edistymistä mahdollisimman pienellä budjetilla.

Seuraavaksi käyn läpi, miksi olemme valinneet kyseiset työkalut ja miten ne auttavat tavoitteessamme luoda käyttäjäystävällinen polku rekisteröinnistä profiiliin luomiseen.

2.4.1 Git ja GitLab

Projektimme versiohallintaa ohjataan Git komennoin. ”Git on DevOps-työkalu, jota käytetään lähdekoodin hallintaan. Se on avoimen lähdekoodin versionhallintajärjestelmä, jota käytetään pienistä hyvin suuriin projekteihin tehokkaasti. Gitin avulla seurataan lähdekoodin muutoksia, jolloin useat kehittäjät voivat työskennellä yhdessä epälineaarisen kehityksen parissa.” (simplilearn 2022) Päädyimme Git versionhallintatyökaluun, sillä se on yleinen, ilmainen käyttää ja tuttu järjestelmä jo projektin alkaessa.

Git lähdekoodin isännöintipalveluna käytämme GitLab-verkkopalvelua. GitLab on palvelin, joka hallinnoi Git-tietovarastoja. Valitsimme projektiin GitLabin, sillä sen avulla koodia voidaan tarkastella, hallita ja kommentoida puolin ja toisin. Isona apuna meillä on ollut GitLabin jatkuvan integraation toiminnallisuus, josta GitLab kertoo seuraavasti: "GitLab auttaa toimitustiimejä ottamaan jatkuvan integroinnin täysin käyttöön ja automatisoimaan koodin rakentamisen, integroinnin ja todentamisen. GitLabin alan johtavat CI-ominaisuudet mahdollistavat automatisoidun testauksen, staattisen analyysin tietoturvatestauksen, dynaamisen analyysin tietoturvatestauksen ja koodin laatuanalyysin, joiden avulla kehittäjät ja testaajat saavat nopeaa palautetta koodinsa laadusta." (Software Freedom Conservancy s.a.)

Kehitämme projektia kehityshaaroissa. Kehitettävän osan valmistuessa se ladataan GitLabiin ja kun haarasta luodaan yhdistämispyyntö (Merge Request = MR), GitLab tarkistaa onko siinä ongelmia liitettävään haaraan nähden. Koodi katselmoidaan kollegan puolesta läpi ja kommentoidaan tarvittaessa. Hyväksynnän jälkeen kyseinen osa yhdistetään version päähaaraan. Päähaaraan päivittäminen käynnistää testiympäristön päivityksen, mikä tuo esiin mahdolliset virheet. Mikäli ongelmia ei synny testiympäristössä, voidaan päivitys viedä tuotantohaaraan. Testiympäristö löytyy osoitteesta <https://www.staging.yourtone.io/fi/>.

2.4.2 Visual Studio Code (VSC)

"Visual Studio Code on virtaviivainen koodieditori, joka tukee kehitystoimintoja, kuten virheenkorjausta, tehtävien suorittamista ja versionhallintaa. Sen tavoitteena on tarjota vain ne työkalut, joita kehittäjä tarvitsee nopeaan koodin rakentamis- ja virheenkorjaussykliin, ja jättää monimutkaisemmat työkulut monipuolisemmille IDE-ohjelmille." (Visual Studio Code s.a.)

Päätimme ottaa Visual Studio Code-editorin käyttöön, sillä siihen saa helposti ladattua työntekoa helpottavia lisäosia, joita voi selata ja etsiä omalta välilehdeltään. Lisäksi VSC kysyy, haluaako käyttäjä ladata lisäosan esimerkiksi koodikielen lukemista tai käyttöä helpottaen, kun ohjelma havaitsee tietyn kielen olevan käytössä. VSC:stä voi myös etsiä sopivia lisäosia, jotka helpottavat muun muassa koodin luettavuuden muokkaamisessa. Vaihtoehtoisia ohjelmia ovat mm. Eclipse tai Brackets. VSC sen sijaan tarjoaa helpon käyttöliittymän lisätä tarpeellisia toimintoja. VSC on visuaalisesti moderni ja sen puolesta helppo käyttää.

2.4.3 Figma

Figma on selaimella toimiva suunnittelutyökalu, jolla voi luoda muun muassa vektorigrafiikkaa, käyttöliittymiä, verkkosivustojen ja mobiilisovelluksien suunnitelmia ja prototyyppejä. Projektissamme olemme käyttäneet Figmaa luodaksemme seuraavia sivuston kanalta olennaisia osia: rautalankamallit (wireframe), sivustopolkujen talletukset sekä sivuston visuaaliset suunnitelmat.

Päädyimme käyttämään Figmaa, sillä se on ilmainen ja toimiva ratkaisu verkkosivujen suunnitteluun ja sitä on mahdollisuus käyttää myös työryhmän kanssa yhdessä. Olemme luoneet Figmalla tähän projektiin rautalankamallit, komponentit kuten painikkeet sekä visuaalisen ulkoasun.

Figmassa on sen käyttäjien tekemiä vapaasti käytettäviä komponentteja, kuten esimerkiksi painikkeet ja pudotusvalikot. Näitä komponentteja on mahdollisuus muokata komponenttina niin että se päivittyy kaikille sivuille joissa komponenttia on käytetty. Figmassa voi myös tarkastella luodun komponentin koodia, jota voi käyttää omassa sovelluksessa.

2.4.4 Poedit

Poedit on helppokäyttöinen käännöstyökalu. Se tarjoaa graafisen työkalun Django:n luomalle tiedostolle, johon funktiolla merkityt tekstit on kerätty. Django erottaa gettext funktiolla käännettävät tekstit koodista.

Projektimme on rakennettu englanniksi ja käännetty suomeksi, Poeditin helppokäyttöisyyden ja älykkäiden ominaisuuksien vuoksi valitsimme sen käytettäväksi projektiimme.

2.4.5 Sentry

”Sentry on palvelu, joka tarkkailee ja korjaa kaatumisia reaaliajassa. Se sisältää API:n tapahtumien lähettämiseen useista eri kielistä, eri sovelluksissa.” (Golden s.a.)

Sentry raportoi käyttämäämme kommunikointiväline Slackiin mikäli sivustolla on ongelma, joka estää palvelun toiminnan.

2.4.6 Google Drive

Google Drive on toimistotyökalu ja pilvipalvelu, jonka avulla voi luoda tiedostoja ja jakaa niitä helposti muiden tarkasteltavaksi, kommentoitavaksi ja muokattavaksi.

Google Drive toimii projektimme tiedostojen tallennuspaikkana. Drive-palveluun tallennetaan projektimme visuaaliset elementit, teksti sisältö, liiketoiminnan asiakirjat, markkinatutkimukset sekä markkinoinnin materiaalit. Tässä projektin osassa olemme käyttäneet Drive-palvelua tekstien säilyttämiseen ja ylläpitoon.

2.4.7 Trello

Trello on ilmainen tehtävä- ja projektinhallinnan työkalu. Trelloon on mahdollista perustaa työryhmiä ja työryhmille erilaisia tauluja. Tauluihin voi laittaa haluamansa määrän otsikkoja (esimerkiksi backlog, todo, doing ja done), joiden alle voi tehdä kortteja. Näin työtehtäviä pystytään kategorisoimaan, priorisoimaan ja sitä kautta määrittämään erilaisia projektin osia tai aiheita.

Käytämme tätä työkalua, sillä Trello tarjoaa käyttäjille myös ilmaisen version, joka vastaa projektimme tarpeita. Käytämme projektissamme Trelloa sivuston kehittämiskohteiden priorisointiin, käyttäjäkokemusten tutkimiseen ja ylläpitoon, hallinnollisiin tehtäviin sekä markkinoinnin suunnitteluun.

2.4.8 Kommunikaatiovälineet

Käytämme tiimin väliseen kommunikointiin useita eri kanavia.

Slack on foorumin kaltainen pikaviestintäohjelma, jonka otimme käyttöön sivuston kehittämistä varten.

Whatsapp on nopea viestintäsovellus, jossa keskustelemme tapaamisista ja käsittelemme akuutit asiat.

Google Meet on videoviestintäpalvelu, jonka välityksellä pidämme palavereita projektiin liittyen.

2.5 Teknologiat

Päädymme seuraaviin projektissa käytettäviin teknologioihin Sérgion kokemuksen pohjalta. Näiden teknologioiden valintaan on vaikuttanut bootstrap-ideologia, joka on yleinen startup yritysten keskuudessa ja jonka keskeinen idea on minimoida kuluja.

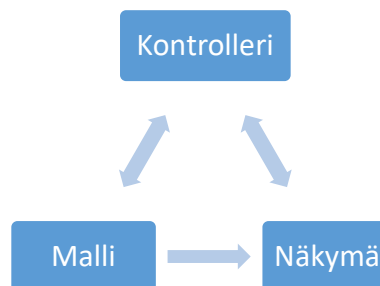
Seuraavaksi käsitellään projektimme keskeisimmät teknologiat ja mitä ne ovat.

2.5.1 Django

”Django on Python-pohjainen verkkosovelluskehys, joka on ilmainen ja avoimen lähdekoodin ohjelmisto. Kehys on yksinkertaisesti kokoelma moduuleja, jotka helpottavat

kehitystä. Ne on ryhmitelty yhteen, ja niiden avulla voit rakentaa sovelluksia tai verkkosivuja sen sijaan, että aloittaisit alusta.” (Simplilearn 2022)

Djangossa käytössä on Model-Template-View-kehys (MTV), joka perustuu mallinäkömääntäjä-kontrolleri-asetelmaan (Lyhyesti MVC eli Model View Controller). Mallielementti (Model) sisältää seuraavan toimintaketjun; kun lähetät sivustolle dataa, malli vastaanottaa sen ja toimittaa sen kontrollerille (Controller), joka puolestaan muuntaa sen tietokantaan sopivaksi sekä säilöo sen. Kontrolleri käsittelee käyttäjän toiminnan ja valitsee näkymän mallin mukaan. Näkymä (View) sisältää käyttöliittymän, jonka sisältö tulee mallielementin kautta.



Kuva 2, Malli, näkymä ja kontrolleri (MVC = Model, View, Controller)

Djangon arkkitehtuuri tukee modulaaristen sovellusten kehittämistä. Sovelluksia pyritään kehittämään funktionaalisen ohjelmistokehityksen kaltaisesti eli siinä rakennetaan itsenäisiä moduuleja, joita voidaan helposti käyttää uudelleen.

Tietokannan rakentaminen ja automatisointi onnistuu helposti Django:n avulla. Malleihin kirjoitetaan tietokannan määrittely ja muutamalla komennolla, tietokannan rakentaminen sekä päivitys sujuvat helposti. Myös lomakkeet, kuten käyttäjän profiilin luonti, voidaan tehdä komennoilla, rakennettujen mallien perusteella. Näkymä, jossa voidaan luoda, päivittää tai poistaa profiilin tietoja, on myös yksinkertaisimmillaan valmiina. Visuaalisesti eriyttä sisältöä Django ei tarjoa, mutta sillä pääsee loistavasti alkuun.

”Django on pääasiassa MTV-kehys (Model-Template-View). Se käyttää terminologiaa Templates for Views ja Views for Controller. Malli liittyy MVC-mallin näkymään, koska se viittaa esitystapaan, joka hallinnoi esityslogiikkaa kehyksessä ja lähinnä ohjaa näytettävää sisältöä ja sitä, miten se näytetään käyttäjälle. Näin ollen Python-koodimme on näkymissä ja malleissa ja HTML-koodimme on templaateissa.” (Django Architecture Data Flair)

Koko projektimme on rakennettu Django kehukseen, jota seuraava teknologia Bootstrap täydentää.

2.5.2 Bootstrap

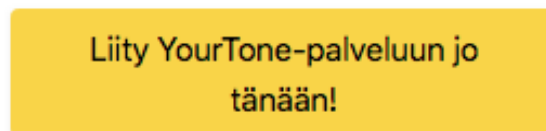
"Bootstrap on kehys, jota käytetään web-sovellusten käyttöliittymän luomiseen. Se tarjoaa css:n, js:n ja muita työkaluja, jotka auttavat luomaan tarvittavan käyttöliittymän.

Djangossa voimme käyttää bootstrapia käyttäjäystävällisempien sovellusten luomiseen."(Django with Bootstrap JavaTpoint)

Komponentteja muokataan luokkamäärittäjätagien sisällä.

Seuraavassa koodiesimerkissä, joka on luotu Bootstrap luokkamäärittäjätagin, määrittäjätagit ovat btn = näppäin, btn-secondary = väri, shadow = varjostus, px-3 = pehmuste horisontaalisti 3 (1-5), mt-3 = marginaali ylös 3 (1-5)

```
<a class="btn btn-secondary shadow px-3 mt-3"
  href="{% url 'account_signup' %}?next={% url 'dashboard:teacher-onboarding' %}">{% trans "Join YourTone today!" %}
</a>
```



Kuva 3, Bootstrap luokkamäärittäjätagin luotu painike.

Bootstrap kompensoi Djangon visuaalisia puutteita, tuo laajan kirjaston valmiita erilaisia komponentteja ja on sen takia hyvä valinta projektimme toteutuksessa.

2.5.3 Python

Projektimme ohjelmointikielenä on Python, sillä Djangon Backend toiminta pohjaa siihen.

"Python on ohjelmointikieli, jonka avulla voit työskennellä nopeasti ja integroida järjestelmiä tehokkaammin." (Python-ohjelmistosäätiö)

Nopeutta Python tarjoaa yksinkertaisella ulkomuodollaan, sillä esimerkiksi verrattuna Javaan siinä on huomattavasti vähemmän kirjoitettavaa sovelluksen luomisessa.

2.5.4 Riippuvuuksien hallinta ja Docker

Aloitimme Pipenv Python-työkalulla, mutta siinä koimme projektinrakennuksen olevan liian hidasta, emmekä keksineet tähän muuta ratkaisua kuin vaihtaa

riippuvuushallintatyökalua. Tämän jälkeen kokeilimme Poetry-työkalua Sérgion ehdotuksesta ja se oli huomattavasti nopeampi ja helpotti työskentelyämme.

Käytin työskennellessä Poetrya lähinnä Dockerin kautta Dockerin buildauksella. Docker tekee kätevän paketin Python sekä käyttöjärjestelmä riippuvuuksista.

”Docker on avoimen lähdekoodin konttialusta. Sen avulla kehittäjät voivat paketoita soveluksia kontteihin - standardoituihin suoritettaviin komponentteihin, jotka yhdistävät soveluksen lähdekoodin käyttöjärjestelmäkirjastoihin ja riippuvuuksiin, joita tarvitaan kyseisen koodin suorittamiseen missä tahansa ympäristössä.” (IBM Cloud Education 2021)

”Kehittäjät voivat luoda kontteja ilman Dockeria, mutta alusta tekee konttien rakentamisesta, käyttöönotosta ja hallinnasta helpompaa, yksinkertaisempaa ja turvallisempaa. Docker on pohjimmiltaan työkalupakki, jonka avulla kehittäjät voivat rakentaa, ottaa käyttöön, ajaa, päivittää ja pysäyttää kontteja yksinkertaisilla komennoilla ja työtä säästävällä automaatiolla yhden API:n kautta.” (IBM Cloud Education 2021)

3 Sovelluksen yleiskuva ja rakenne

Djangon tavoitteena on luoda selkeä kansiorakenne, jota voidaan hyödyntää sovelluksen eri osien luomisessa. Seuraavaksi käydään läpi sovelluksen rakentamisen perusteita ja sitä, miten Django sovellukseen lisätään moduuleja.

3.1 Rakenne ja arkkitehtuuri

Projektin rakenne ja arkkitehtuuri perustuvat Djangon Model-Template-View kehykseen. Sovelluksen osat pyritään pitämään yksittäisinä moduuleina, joita voidaan uudelleen käyttää helposti muissa projekteissa. Rakenne on lyhyesti projektikansio; joka sisältää konfigurointi tiedostot; pääkansio sisältää yksittäiset moduulit ja käyttäjänäkymän (frontend), sekä alimpana ohjelmien kansiot, joissa on ohjelmien toiminta logiikka, omat html sivut ja tietokanta logiikka.

3.2 Sovelluksen osien koodaus

Ohjelman koodauksen ensimmäinen askel on pohtia, voidaanko koodia käyttää muissa projekteissa vai onko se selkeästi vain tähän projektiin liittyvä. Mikäli koodattava palanen on vain kyseiseen projektiin liittyvä, rakennetaan se ohjelmistojen rykelmään (tässä projektissa dashboard). Jos se on käytettävissä sellaisenaan tai pienellä muuntelulla muualla, luodaan sille oma kansio pääkansioon. Itsenäiset ohjelmat lisätään projektin settings tiedostoon apps listaan ja niille konfiguroidaan oma reitti urls tiedostoihin.

3.3 Sovelluksen tietoturvallisuus

Iso osa sovelluksen tietoturvasta rakentuu Django all auth -kirjastoon, jolla on rakennettu kirjautuminen sähköpostilla ja salasanalla, sekä uuden salasanan luominen. Tämän lisäksi käytössä on Oauth-tekniikka, joka antaa mahdollisuuden Facebook- ja Google-kirjautumiseen.

4 Kehittämistyö

Projektin lähdekoodi kopioidaan GitLabista omalle tietokoneelle. Sen jälkeen tarvittavat riippuvuudet saadaan ladattua. Dockerin avulla rakennamme ja haemme projektin tarvittavat riippuvuudet compose build komennolla. Kun build on valmis voimme suorittaa compose run komennon ja paikallinen YourTone verkkosovellus käynnistyy, mikäli ongelmia ei havaita.

4.1 Projektin alku

Projektin pohjan on luonut Sérgio ja siihen sisältyi projektin alustus Django kehiksen avulla sekä konfigurointi GitLab:iin, testiympäristöön (<https://www.staging.yourtone.io/fi/teachers/search/>) ja tuotantoympäristöön (<https://www.yourtone.fi/fi/>)

4.2 Projektiin tehtyjä töitä

Itse olen tehnyt projektiin 37 yhdistämispyyntöä (MR=merge request), joista jokainen on oma haaransa. Näihin haaroihin olen tehnyt tarvittavat muutokset, jotka sitten on tarkistettu (peer reviewed) ja yhdistetty päähaaraan.

Ensimmäisenä tehtävänä asetustiedostoon vaihdettiin aikavyöhykkeeksi Helsinki, sen hetkiset mallit vietiin admin-sivustolle, jotta niiden muokkaaminen admin-käyttöliittymässä oli mahdollista ja määritys lisättiin malli tiedostoon. Kyseiset mallit olivat siis käyttäjän models.py tiedosto, missä määritellään tietokanta tyytit ja mitä voidaan tietokannan lisäksi käyttää myös yksinkertaisten lomakkeiden rakentamiseen.

```

▼ cief/users/models.py
1  - from django.utils.translation import gettext_lazy as _
2  + import uuid
3  - from django.contrib.auth.models import AbstractUser
4  - from django.urls import reverse
5  + from django.db import models
6  + from django.urls import reverse
7  + from django.utils.translation import gettext_lazy as _
8
9  - class User(AbstractUser):
10 -     """User Model"""
11 -     username = None
12 +     username = models.CharField(
13 +         _('username'),
14 +         max_length=150,
15 +         unique=True,
16 +         default=uuid.uuid4
17 +     )
18
19     email = models.EmailField(_('email address'), unique=True)
20
21     USERNAME_FIELD = 'email'
22     REQUIRED_FIELDS = []
23
24     REQUIRED_FIELDS = ['username']
25
26     def get_absolute_url(self):
27         """Get url for user's detail view.

```

Kuva 4, käyttäjä mallin luominen

Otimme Mailhog-palvelun tilapäisesti käyttöön, jotta sähköposti ominaisuuksia pystyttiin testaamaan.

Rekisteröinnin loppuun lisättiin alkuperäisessä versiossa kolme vaihtoehtoa, etsi opettajaa, rekisteröidy opettajaksi ja rekisteröidy artistina. Myöhemmin täsmensimme markkinasegmenttiä ja artisti jäi pois.

```

clef/dashboard/templates/signupredirect/signupredirect.html 0 → 100644 +19 -0 Viewed
1 + {% extends "account/base.html" %}
2 +
3 + {% load i18n %}
4 +
5 + {% block head_title %}{% trans "Start" %}{% endblock %}
6 +
7 + {% block inner %}
8 + <h1>{% trans "Choose your type" %}</h1>
9 +
10 + <p>{% trans 'What would you like to do?' %}</p>
11 + <p>{% trans 'Also please note, that none of the choices are permanent or exclusive. But to register as a Teacher, an Artist or to schedule a
meeting, verification is required. Anyone can search and add people to favorites.' %}</p>
12 +
13 + <div class="list-group" >
14 +   <a href="/" class="list-group-item list-group-item-action">Search a Teacher</a>
15 +   <a href="/dashboard" class="list-group-item list-group-item-action">Register as a Teacher</a>
16 +   <a href="/dashboard" class="list-group-item list-group-item-action">Register as an Artist</a>
17 + </div>
18 +
19 + {% endblock %}

```

Kuva 5.1, templaatti joka jatkaa base templaattia.

```

clef/dashboard/urls.py +6 -1 Viewed
1 1 from django.urls import path
2 2
3 - from .views import DashboardView
3 + from .views import DashboardView, SignupredirectView
4 4
5 5 app_name = "dashboard"
6 6
@@ -10,4 +10,9 @@ urlpatterns = [
10 10     view=DashboardView.as_view(),
11 11     name="dashboard"
12 12     ),
13 +     path(
14 +         "signupredirect",
15 +         view=SignupredirectView.as_view(),
16 +         name="signupredirect"
17 +     )
13 18 ]
\ No newline at end of file

clef/dashboard/views.py +8 -0 Viewed
@@ -9,3 +9,11 @@ class DashboardView(LoginRequiredMixin, TemplateView):
9 9     def get_context_data(self, **kwargs):
10 10         context = super().get_context_data(**kwargs)
11 11         return context
12 +
13 + class SignupredirectView( TemplateView ):
14 +
15 +     template_name = "signupredirect/signupredirect.html"
16 +
17 +     def get_context_data(self, **kwargs):
18 +         context = super().get_context_data(**kwargs)
19 +         return context

```

Kuva 5.2, uudelleenohjaamisen asetukset dashboardin url-tiedostoon, sekä views eli kontrolleri tiedostoon

Toisinaan Sérgio myös mentoroi tekemistäni, jotta ymmärryksen Django MTV malliin syventyi. Rakensimme yhdessä käyttäjämallit loppuun.

Rekisteröinti tapahtui alkuun Django generisillä lomakkeilla. Mutta työstin ensiksi profiilin asetussivua, mistä pystyi näkemään omat tiedot, sekä pääsi muokkauslomakkeille.

YourTone

Personal information

TODO
 First Name: Bob
 Last Name: McFrog
 Email: admin@tester.com

[Edit information](#) [Email settings](#)

Teacher profile

TODO: Profile picture, demo video, verifications
 Your base price: €5.00

Short description is shown in the teacher list
 Short description: A short description

Long description is shown when your profile is selected
 Long description: A somewhat longer description

[Edit teacher profile](#)

Kuva 6, varhainen profiili sivu, ennen julkaisua.

Opettajan tietoihin olennaista on lisätä, liikkuuko opettaja oppilaan luokse vai toisinpäin, templateen loin ehtolausekkeen tämän esittämistä varten.

```

53 - {% trans "Will you teach from home?" %} "{{teacher_profile.student_travels_to_teacher}}", {% trans "Will you travel to student" %} "
    {{teacher_profile.teacher_travels_to_student}}"
53 + {% trans "Will you teach from home?" %} =-{{if teacher_profile.student_travels_to_teacher}}{% trans "Yes"%}{%else%}{% trans "No"%}
    {%endif%}, {% trans "Will you travel to student?" %} =-{{if teacher_profile.teacher_travels_to_student}}{% trans "Yes"%}{%else%}{% trans "No"%}
    {%endif%}

```

Kuva 7, valintalausekkeen luominen templaatissa.

Ilman muokkausta Django lomakkeilla ei kovin pitkälle pääse, mutta muokkaus on hyvin yksinkertaista. Tässä kuvassa olen määritellyt opettajan tietojen käsittelyä ja miten lomake lukee niitä. Kuvasta näkee, että Djangoille kerrotaan mihin malliin lomake perustuu (model = TeacherProfile), sekä miten määritellään kentät, lisäosat, aputekstit ja pakolliset kentät.

```

clef/dashboard/views/forms.py 0 → 100644
2 + from teachers.models import TeacherProfile
3 + from django import forms
4 + from datetime import datetime
5 +
6 + year = datetime.now().year
7 + birth_year_choices = list(range(year, year - 120, -1))
8 +
9 + class TeacherForms(forms.ModelForm):
10 +
11 +     def __init__(self, *args, **kwargs):
12 +         super().__init__(*args, **kwargs)
13 +
14 +         for field in self.Meta.required:
15 +             self.fields[field].required = False
16 +
17 +     class Meta:
18 +
19 +         model = TeacherProfile
20 +
21 +         fields = [
22 +             'birthdate', 'gender',
23 +             'description', 'short_description',
24 +             'student_travels_to_teacher', 'teacher_travels_to_student',
25 +             'address', 'city', 'state', 'postal_code', 'country',
26 +             'base_price', 'facebook', 'soundcloud', 'youtube', 'instagram'
27 +         ]
28 +
29 +         localized_fields = ('__all__')
30 +
31 +         widgets = {
32 +             'birthdate': forms.SelectDateWidget(
33 +                 years=birth_year_choices,
34 +             )
35 +
36 +         help_texts = {
37 +             'birthdate': _('Students have option to filter teachers by age.'),
38 +             'gender': _('Students have option to filter teachers by gender.'),
39 +             'description': _('Description is limited to x characters, students who visit your profile can read this.'),
40 +             'short_description': _('Short description is limited to x characters, it is shown in the list where students can browse teachers.'),
41 +             'address': _('You need an address to show up in results, teachers are arranged by proximity to the student.'),
42 +             'facebook': _('Social media accounts give you credibility.'),
43 +         }
44 +
45 +         required = (
46 +             'facebook', 'soundcloud', 'youtube', 'instagram'
47 +         )

```

Kuva 8, lomakkeen hienosäätöä käsin.

Muokatun lomakkeen käyttöönotto kerrotaan myös kontrollerille, joka ohjaa opettajanäkymien toimintaa, kuten seuraavassa kuvassa yhdeksän näemme.

The screenshot shows a code editor window with the following Python code:

```

1 1 from django.contrib.auth import models
2 2 + from django.contrib.auth.mixins import LoginRequiredMixin
3 3 from django.views.generic import UpdateView, CreateView
4 4 from teachers.models import TeacherProfile
5 5 from django.urls import reverse
6 6 + from .forms import TeacherForms
7 7
8 8
9 9 - class EditTeacherProfileView(UpdateView):
10 10 + class EditTeacherProfileView(UpdateView, LoginRequiredMixin):
11 11
12 12
13 13
14 14
15 15
16 16 + def form_valid(self, form):
17 17 +     form.instance.user = self.request.user
18 18 +     return super().form_valid(form)
19 19 +
20 20
21 21
22 22
23 23
24 24
25 25
26 26
27 27 - class CreateTeacherProfileView(CreateView):

```

Below the code, there is a GitHub comment by Sérgio Isidoro (@smaisidoro) from 1 year ago, resolved by Niko Hautala. The comment says: "I had completely missed this. Good job!". There is a thumbs up reaction and a "Reply..." input field.

Kuva 9, kommentointi ja koodin tarkistus on tärkeä osa toimivan tuotteen rakennusta.

Tehtäviin kuului paljon myös lokalisaation kanssa työskentelyä ja Django komennolla luotua käännöstiedostoa oli helppo muokata Poedit-sovelluksella.

Koodi, sekä alkuperäiset tekstit ovat kaikki englanniksi, ne on tägien sisällä määritelty `{% trans "text" %}` joka kertoo Django automaatti lokalisaatio työkalulle, make messages, että tuo teksti lisätään tiedostoon `django.po`. Tällainen tiedosto tehdään jokaista kieli-asetusta kohden. Lyhyesti se toimii niin, että koodissa käytetty `{% trans "text" %}` toimii id arvona, jolle annetaan käännös vastapariksi.

4.3 Vaatimusmäärittely

Vaatimusmäärittelyä on tehty tiiminä, sen perustana on markkinatutkimus, jonka avulla olemme kartoittaneet mitä tarvitsemme markkinoiden saavuttamiseen.

Siihen sisältyi käyttäjäkyselyt kumpaankin asiakasryhmään, kilpailija-analyysi ja konsultaatio useista lähteistä.

4.4 Responsiivisuus

Responsiivisuus rakentuu useammasta palasta, iso osa siitä on Bootstrap kirjaston kautta, mutta esimerkiksi fontti on rakennettu yhtälöllä, joka käyttää hyväkseen `calc()` funktiota ja määrittelee minimi, sekä maksimi koot ja muovautuu kuvaruudun koon mukaan siltä väliltä. `font-size: calc(30px + (24 - 16) * (100vw - 400px) / 1200);`

4.5 Monikielisyys

" Jotta Django-projekti olisi käännettävissä, sinun on lisättävä vähimmäismäärä koukkuja Python-koodiin ja malleihin. Näitä koukkuja kutsutaan käänösmerkkijonoiksi. Ne kertovat Djangolle: "Tämä teksti on käännettävä loppukäyttäjän kielelle, jos tekstille on saatavilla käänös kyseisellä kielellä.". Käännettävien merkkijonojen merkitseminen on sinun vastuullasi; järjestelmä voi kääntää vain merkkijonoja, joista se tietää." (Django-ohjelmistosäätiö s.a.)

Django ominaisuuksin kuuluu `gettext()` funktio, joka kertoo Djangolle kyseisen tekstin kuuluvan käännettäviin. `Manage.py` hallintatiedoston komennolla tee viestit (`makemessages`) Django rakentaa lokalisoitun version käännettäviksi merkatuista teksteistä. Tiedosto on yksinkertaisinta lukea Poedit sovelluksella, joka tarjoaa helpon ja ymmärrettävän käyttöliittymän lokalisaatio tiedoston tarkasteluun.

```
#: pages/templates/pages/for_teachers.html:38
msgid ""
"YourTone is a search engine for private music lessons and is designed to "
"make learning music and finding teachers easier than ever before."
msgstr ""
"YourTone on musiikin yksityisopetuksen hakukone ja sen tarkoitus on tehdä "
"musiikin oppimisesta ja opettajien etsimisestä helpompaa kuin koskaan ennen."
```

5 Pohdinta

Palvelu, asenne, oppi ja tavoitteet. Opinnäytetyö tarjosi tekijälle paljon pohdinnan aiheita monipuolisesti työhön, yritykseen ja itseen liittyen.

5.1 Palvelu

Palvelu YourTone käynnistyi, joskin hieman myöhässä, toteutusta hidasti tiimiläisten elämäntilanteet ja muut työt. Tämä Startup projekti oli muille kuin Sérgiolle täysin uusi kokemus ja monessa osa-alueessa on opittu valtavasti, niin markkinoinnissa kuin palvelun rakentamisessa. Suurimpia alun virheitä oli epätietoisuus minimi toimivasta tuotteesta (MVP = Minimum Viable Product) ja jo 2022 julkaisemamme ensimmäinen versio oli huomattavasti hiotumpi kuin oli tarpeen. Lisäksi palvelun käyttäjien hankinta alkoi kömpelösti. Meillä oli maksumenetelmät heti käytössä, mutta uuden palvelun julkaisu olisi vaatinut myöhemmin keksimämme koukun eli maksat vasta kun saat palvelumme kautta kontaktin, jo alusta asti.

5.2 Asenne

Itse opinnäytetyön teko YourTone projektin yhteydessä oli myös kömpelöä ja sitä vaikeutti alituinen epävarmuus tekijän ammattitaidosta, niin opinnäytetyön, kuin ohjelmoinnin kanssa. Ohjauksen avulla saatiin määriteltyä tarkka rajaus opinnäytetyölle, mutta työllistyminen laittoi opinnäytetyön tauolle. Tämän tauon aikana rajattu alue tuli tekijän kollegan tekemänä valmiiksi. Seuraavan vuoden aikana tekijä suoritti pienkehitystä ja korjausta YourTone projektille, joita tässä opinnäytetyössä on käyty läpi.

Opinnäytetyö oli huomattavan haastava aihe ja tekijän henkinen tila ei riittänyt huomattavan ison stressin hallintaan, jota lähinnä aiheutti heikko taloudellinen tilanne.

Työtä olisi helpottanut huomattavasti, jos sille olisi antanut aikaa ja käynyt läpi opinnäytetyöpajan ja muita työn helpottamiseen annettuja työkaluja. Myös kommentointi ja konsultointi opettajien, muiden oppilaiden, tuttujen ja lähisukulaisten kanssa olisi auttanut, mutta jostain syystä tekijä oli varsin yksin työn kanssa.

Suuret henkiset esteet saivat tekemisen hyvin vaivaloiseksi. Lopulta kuitenkin viime hetkellä, onnistui ajatus käännättää ja työnteko muuttui merkittävästi helpommaksi.

5.3 Oppi

Opinnäytetyön teko ja oppiminen olivat hyvin haasteellisia. Tuntemattomien asioiden epäjärjestelmällinen haltuunotto sai työn vaikuttamaan heti alkuun toivottomalta, niin ohjelmistoinnin, kuin myös opinnäytetyön kirjoittamisen kannalta. YourTone-Tiimillä oli haasteita työalueiden määrittelyssä, sekä oli myös luonnollista, että pienellä ryhmällä tehtävät olivat monipuolisesti kaikkien hartioilla. Tekijä opetteli ja työskenteli monien uusien ja haastavien asioiden parissa, kuten opinnäytetyö, startup yrittäjyys, sekä uudet teknologiat ja ohjelmointi pienessä ryhmässä ilman selkeitä työnkuvia tai työaikoja.

Tekijä oli jo harjoittanut tehtävähallintaa Trelion avulla koulun aikana, mutta jostain syystä se jäi opinnäytetyön kanssa tekemättä, suurelta osin varmasti epätoivon ja stressin tunteet ohjasivat toimintaa.

5.4 Tavoitteet

Miten luoda käyttäjäystävällinen polku rekisteröinnistä profiiliin luomiseen.

Tekijä rakensi rajaukseen määritellyn alueen osaksi, työllistymisen jälkeen Sérgio teki tarvittavan alueen palvelun julkaisua varten. Myöhemmin alueen rajausta on työstetty eteenpäin niin mekaniikoilta, kuin visuaalisesti ja se löytyy opinnäytetyön liitteenä kolmessa sivussa. Tekijä työskenteli laajalti yrityksen asioiden kanssa, tukien toimintaa pienkehitystyön, sekä yritystoiminnan kehityksen kanssa. Yritystoiminta jatkuu edelleen.

Lähteet

Data Flair. Django Architecture – 3 Major Components of MVC Pattern. Luettavissa: <https://data-flair.training/blogs/django-architecture/>. Luettu 16.11.2022

Django-ohjelmistosäätö. Django dokumentaatio. Luettavissa: <https://www.djangoproject.com/>. Luettu 21.11.2021

Django-ohjelmistosäätö. Django käännökset. Luettavissa: <https://docs.djangoproject.com/en/4.1/topics/i18n/translation/>. Luettu 4.5.2022

Golden. Sentry (software company). Luettavissa: [https://golden.com/wiki/Sentry_\(software_company\)-EAAMVBX](https://golden.com/wiki/Sentry_(software_company)-EAAMVBX). Luettu 21.11.2022

IBM Cloud Education 23.06.2021. Docker. Luettavissa: <https://www.ibm.com/en/cloud/learn/docker>. Luettu 21.11.2022

JavaTpoint. Django with Bootstrap. Luettavissa: <https://www.javatpoint.com/django-and-bootstrap>. Luettu 21.11.2022

Microsoft 2022. Visual Studio Code. Luettavissa: <https://code.visualstudio.com/docs/supporting/FAQ>. Luettu 22.11.2022

Python-ohjelmistosäätö. Python dokumentaatio. Luettavissa: <https://www.python.org/>. Luettu 21.11.2022

Simplilearn Solutions 2022. What is Git: Features, Command and Workflow in Git. Luettavissa: <https://www.simplilearn.com/tutorials/git-tutorial/what-is-git>. Luettu 22.11.2022

Simplilearn Solutions 2022. All You Need To Know About Django Framework. Luettavissa: <https://www.simplilearn.com/tutorials/django-tutorial/what-is-django-python>. Luettu 22.11.2022

Software Freedom Conservancy. GitLab dokumentaatio. Luettavissa: <https://about.gitlab.com/>. Luettu 21.11.2022

Liitteet

Liite 1. Rekisteröityminen sivustolle

3. Sivua

Liite 1. Rekisteröityminen sivustolle 1/3



Etsi Opettajille Hinnoittelu YourTone tiimi Ota yhteyttä

Rekisteröidy Kirjautu sisään

Rekisteröidy

Onko sinulla jo tili? Ole hyvä ja [Kirjautu](#).

Rekisteröidy sähköpostilla ja salasanalla

Nimi*

Syntymäpäivä*

Syntymäaika vaaditaan sekä oppilailta että opettajilta

Sähköposti*

Salasana*

Salasana toistamiseen*

Hyväksyn palvelun ehdot*

Tilaa uutiskirje ja vastaanota uutisia ja tarjouksia

Rekisteröidy >

Rekisteröidy yhteys tilillä

Facebook

Google

[Opettajille](#) [Ehdot ja edellytykset](#)

[Hinnoittelu](#) [Tietosuojakäytäntö](#)

[YourTone tiimi](#)

[Ota yhteyttä](#)



Kieli

suomi ▾

Liite 1. Rekisteröityminen sivustolle 2/3



Etsi Opettajille Hinnoittelu YourTone tiimi Ota yhteyttä

MyTone Kirjautu ulos

Niko Hautala

Vahvistusviesti on lähetetty osoitteeseen niko.hautala@myy.haaga-helia.fi.



Tervetuloa YourTone-sivustolle!

Valmistellaan tilisi

Tulin oppimaan

Etsin opettajaa, joka auttaisi minua parantamaan musiikillista osaamistani. YourTone-oppilastilin avulla voin tallentaa suosikki opettajia, lähettää viestejä ja varata oppitunteja.

Jatka oppilaana

Olen tullut opettamaan

Olen ammatti- tai harrastemuusikko, joka etsii opiskelijoita. YourTone-tilin avulla voin luoda profiilin musiikillisesta osaamisestani ja oppilaat voivat löytää minut YourTonessa ja muissa hakukoneissa

Jatka opettajana

Tämä valinta ei ole lopullinen. Voit myöhemmin luoda opettajaprofiilin, jos valitset nyt opiskelijan.

[Opettajille](#) [Ehdot ja edellytykset](#)

[Hinnoittelu](#) [Tietosuojakäytäntö](#)

[YourTone tiimi](#)

[Ota yhteyttä](#)



Kieli

suomi

Liite 1. Rekisteröityminen sivustolle 3/3

Henkilökohtaiset tiedot



Etunimi: Niko

Sukunimi: Hautala

Sähköposti: niko.hautala@myy.haaga-helia.fi

Sukupuoli :

Syntymäpäivä: 26. maaliskuuta 1989

Muokkaa tietoja

Vaihda salasana

Sähköpostiasetukset

Vaihda kuvasi

Yhdistä yhteys tili kirjautuaksesi

Luo opettajan profiili



Kieli

suomi ▾