**Harjoitustyöraportti**

**Web-Projekti 2**

Janne Hyyryläinen

Tiia Itkonen

Riku Kalliokoski

Markus Moilanen

Oskari Sutela

Harjoitustyöraportti

Huhtikuu 2017

Tekniikan ja liikenteen ala

Insinööri (AMK), Tieto- ja viestintätekniikka

Mediatekniikka

**Sisältö**

[1 Johdanto 1](#_Toc479318082)

[2 Projektiyhmä 2](#_Toc479318083)

[3 Sovelluksen kokonaisrakenne 2](#_Toc479318084)

[4 Tärkeimmät Funktiot (ja luokat) ja välitettävät parametrit 2](#_Toc479318085)

[5 Sovelluksessa hyödynnetyt tietovarastot 2](#_Toc479318086)

[5.1 Tietokanta 2](#_Toc479318087)

[Tiedostot 2](#_Toc479318088)

[Ulkoiset tietolähteet? 2](#_Toc479318089)

[6 Lähdekoodit 2](#_Toc479318090)

[7 Resurssit 3](#_Toc479318091)

[8 Itsearviot 3](#_Toc479318092)

[Ryhmän yhteinen itsearvio 3](#_Toc479318093)

[Jannen itsearvio 3](#_Toc479318094)

[Tiian itsearvio 3](#_Toc479318095)

[Rikun itsearvio 3](#_Toc479318096)

[Osakarin itsearvio 3](#_Toc479318097)

# Johdanto

Aika- ja persoonamuodot (passiivi)

Tarkistakaa puolipisteen käyttö, jos käytätte

Voi laittaa kuvia

Tässä dokumentissa kuvataan toteutetun projektin taustaa, tehtäviä, resursseja ja projektiryhmää. Projekti toteutettiin osana Jyväskylän ammattikorkeakoulun Web-Projekti 2 TTMS0700-opintojaksoa keväällä 2017.

Projektin tavoitteena oli päivittää syksyllä suoritettujen Web-palvelun määrittely ja suunnittelu -kurssien pohjalta luotua ja suunniteltua verkkokauppaa paremmaksi, lisäämällä siihen tietokantatoteutus. Samalla verkkokaupan visuaalista ilmettä parannetaan ja web-sovellusta uudistetaan niin, että se hyödyntää enemmän asiakas- ja palvelinpuolen tekniikoita, joita hyödynnettiin kevään Web-palvelinohjelmointi -kurssilla.

# Projektiryhmä

Projektiryhmään kuuluvat viisi Jyväskylän ammattikorkeakoulun ICT-opiskelijaa: Janne Hyyryläinen, Tiia Itkonen, Riku Kalliokoski, Markus Moilanen ja Oskari Sutela. Projektin front-end -puoli jaettiin Itkosen ja Sutelan vastuulle. Back-endin toteutuksesta vastaavat Hyyryläinen, Kalliokoski ja Moilanen. Jokaiselle projektin jäsenelle haluttiin kuitenkin taata mahdollisuus tehdä jokaista osa-aluetta.

* Käytännön toteutuksen selostus
* tähän vois varmaan myös kertoa kommunikaatiosta (trello, slack, whatsapp)

# Sovelluksen kokonaisrakenne

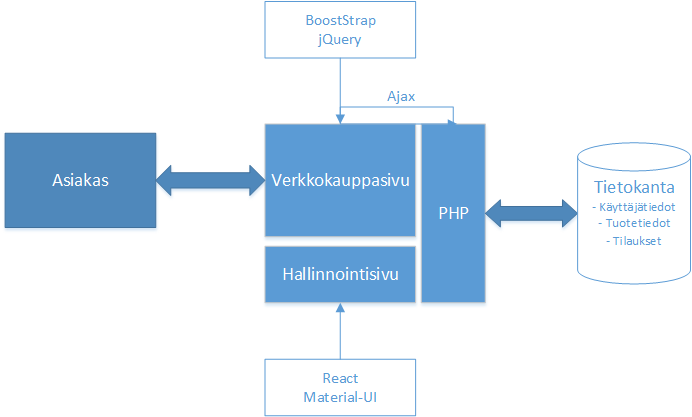
Käytännössä: mistä osista koostuu, miten osat kommunikoi keskenään? apuna voi käyttää alla olevaa kuvaa

Verkkokaupan visuaalinen ilme toteutettiin perus-HTML- ja CSS-tekniikoilla. Layoutin luonnissa käytettiin BootStrapin gridiä ja tyyliluokkia. Myös navigointipalkin pohja tulee BootStrapistä.

Avatkaa lisää ^

JavaScriptiä käytettiin sivulla muutamiin toimintoihin, kuten tuotenäkymän näyttämiseen, mutta suurimmaksi osaksi Ajax-funktioiden ajamiseksi. Ajaxilla otetaan yhteys PHP-tiedostoihin, jotka taas ovat yhteydessä tietokantaan.

Erillisellä web-sivulla sijaitsevalla hallinnointisivulla admin-käyttäjät voivat muokata tietokannan tietoja eri tavoin. React-frameworkilla toteutetulla sivulla on erilliset sivut tuotteiden ja käyttäjien lisäämiselle sekä tuotteiden, käyttäjien ja tilausten listaamiselle. Tarvittavia jsx-elementtejä otettiin käyttöön muun muassa Material-UI-kirjastosta ja muualta netistä.



Kuva 1. Sovelluksessa käytetyt tekniikat, frameworkit ja tietokannan pääsisältö

Varsinaisia luokkarakenteita ei käytetty ohjelmoinnin tukena, paitsi React-luokkia.

# Tärkeimmät funktiot ja välitettävät parametrit

ainakin tärkeimpien toiminta dokumentoitava, välitettävät parametrit

Hienoimmat tekniset highlightit

Asiakaspuolen JavaScript-koodissa hyödynnettiin paljon jQueryn Ajax-metodia. Post-kutsujen parametreina lähetettiin muun muassa haku- ja lajitteluehtoja ja tuote-ID:itä. Ajax ottaa yhteyden erilaisiin PHP-tiedostoihin, jotka suorittavat jonkin MySQL-tietokannan toiminnon, esimerkiksi haun tai lisäyksen. Sen jälkeen ne palauttavat tarvittavat tiedot, kuten tuotetiedot.

Kirjautumiseen, rekisteröitymiseen ja tuotteen lisäämiseen käytettiin erilaisia lomakkeita. Lomakkeiden kentät tarkastettiin käyttäen regular expressionia, ettei käyttäjä voi lisätä tietokantaan ihan mitä tahansa tietoja. Kenttien sallitut merkit ja pituudet käydään läpi ja Ajax-funktio suoritetaan vasta, kun kaikki kentät läpäisevät testin.

Suolaus ja muuta jos haluaa

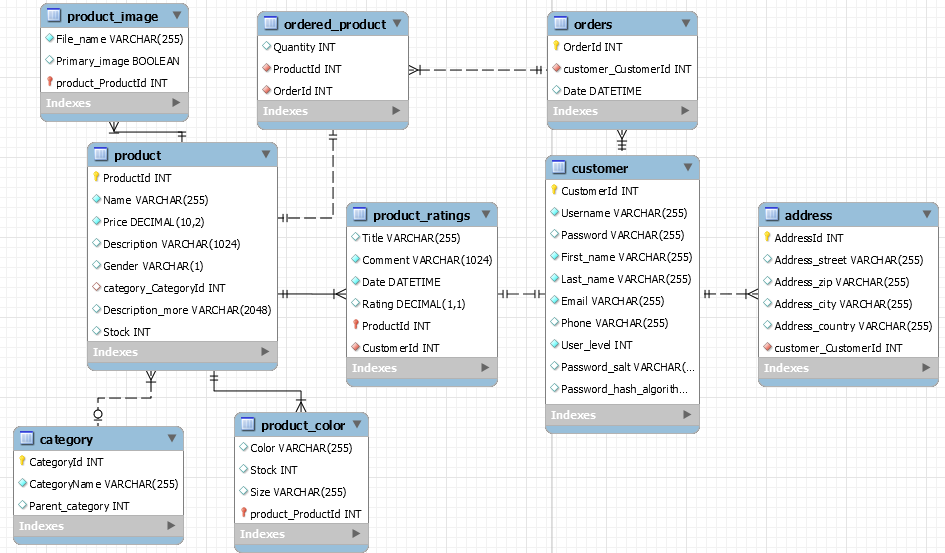
# Sovelluksessa hyödynnetyt tietovarastot

## Tietokanta

Projektia varten luotiin MySQL-tietokanta Labranettiin. Sinne tallennettiin käyttäjiin, tuotteisiin ja tilauksiin liittyvät tiedot. Käyttäjille tallennettiin kirjautumis- ja osoitetiedot. Tuotteille taas tallennettiin kaikki tarvittavat mitä näytetään sivulla, kuten nimi, hinta, kuvat ja tuotetiedot. Lisäksi tuotteille voidaan lisätä kommentteja, jotka tallennetaan tietokantaa. Myös tilausten tiedot tallennetaan oston yhteydessä.

Hallinnointi-sivun yhteydessä käytettiin paljon tietokantaa. Tietoja tulostetaan näkyville admin-käyttäjälle ja erillisillä sivuilla voidaan myös lisätä dataa. Verkkokaupan puolella käyttäjät voivat luonnollisesti muokata omia käyttäjätietojaan. Tuotelistauksessa haetaan kaikki tuotetiedot kannasta.

Tietokannan kanssa kommunikointiin käytettiin paljon erilaisia PHP-tiedostoja, jotka ajavat kantaan SQL-lausekkeita liittyen lisäykseen, poistoon, muokkaukseen, järjestämiseen. Join-lausekkeilla yhdistettiin eri taulujen tietoja.

Tietokannan suunnitteluvaiheessa otettiin huomioon mahdollisia tulevaisuuden toimintoja, kuten valmistajat tuotteille sekä alennukset tai kampanjat. Niitä ei kuitenkaan otettu käyttöön valmiissa projektissa. Suunnittelu tehtiin MySQL Workbenchin mallintajalla.   


## Tiedostot

Php-tiedostot jakautuvat seuraaviin kategorioihin:

* Kirjautumiseen, tilin muokkaamiseen ja rekisteröintiin liittyvät tiedostot
* Verkkokauppasivu ja popup-sivujen tiedostot
* Toiminnallisiin osuuksiin (kommentointi, ostoskori)

Javascript-tiedostot jakautuvat seuraaviin kategorioihin:

* Jsx –elementtitiedostot, joista React-osuus koostuu
* Käyttäjätietojen ohjaamiseen liittyvät tiedostot
* Kommentoinnin siirtoon liittyvä ohjeistustiedostot

# Lähdekoodit

Sovelluksen lähdekoodit löytyvät kokonaisuudessaa githubista \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ja studentinpalvelimelta osoitteesta: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Resurssit ja Itsearviot:

Selvitys työhön käytetystä ajasta ja tekijöiden roolista harjoitustyön tekemisessä sekä työn jakaantumisesta ryhmän sisällä.

Lähtökohtaisesti projektiin on varattu 81 tuntia per henkilö. Projektiryhmän resurssit jakautuivat näin:

## Janne:

Tunteja yhteensä: h

Tekemiset:

* Ostoskori
* Tilinhallinta (muokkaus)
* Salasanojen suolaus
* Tuotehaku
* Auttaminen rekisteröinnissä ja kirjautumisessa
* Navbar II
* Loginin liitto navbariin

Itsearvio:

## Riku:

Tunteja Yhteensä:

Tekemiset:

* Tunteja yhteensä:
* Tekemiset:
* Ympäristönluontia
* Rekisteröinti
* Kirjautuminen
* Regex
* Dokumentointi
* Toiminnallisuuksien ja pohja yhdistelyä
* Muotoiluja ja korjauksia tyylittelyyn ja toiminnallisuuteen
* Katselmointia

Itsearvio:

## Tiia:

Tunteja yhteensä:

Tekemiset:

* Kommentointi
* Yhteensovittamista
* tyylittelyä

Itsearvio: Onnistumiset, puutteet, huomioita

## Markus:

Tunteja yhteensä: ~75h

Tekemiset:

* Tietokanta
* Hallinnointi-sivu
* Back-end tuki
* Dokumentointi

Itsearvio: Suuri osa ajasta meni Reactin ja Webpackin säätämisessä. Ilman uuden frameworkin opettelua olisi saanut ehkä aikaan enemmän tai ainakin tehokkaammin, mutta toisaalta sen opettelu oli itselle hyödyllistä. Hallinnointisivu onnistui kohtalaisesti ja tarvittavat ominaisuudet sisältävänä, vaikka koodi onkin rumaa. Olisin voinut myös kommentoida paremmin. Koulun koneella ei voinut tehdä Reactiä, joten autoin tapaamisissa muita ryhmäläisiä koodin kanssa. Kurssin alussa tein tietokannan, jota käytettiin laajasti projektissa.

## Oskari

Tunteja yhteensä:

Tekemiset:

* Pop-up ja verkkokauppa luonti
* Visualisointi

Itsearvio: Onnistumiset, puutteet, huomioita

Projektiryhmä:

Tunteja yhteensä:

Itsearvio: Onnistumiset, puutteet, huomioita