

|  |
| --- |
| Antero Päivärinta |
| Toimitsijoiden hallinta - case Kalevan kisat |

|  |
| --- |
| Yksikkö (Liiketalous, Sosiaali- ja terveysala, Tekniikka koko 14) tai Hyvinvoinnin asiantuntija YAMK / Teollisuusrobotiikka YAMK  Liiketalous / Tekniikka / Sosiaali- ja terveysala |
| 202X |

VAASAN AMMATTIKORKEAKOULU (koko 12)

Koulutuksen nimi (esim. Sähkötekniikka/ Liiketalous/ Hoitotyö, fontti koko 12). Huomaa, että koulutuksen nimi tulee aina koulutuksen kielellä myös englanninkielisessä abstraktissa.)

**TIIVISTELMÄ (koko 14)**

Tekijä owner

Opinnäytetyön nimi Opinnäytetyön nimi :

Mahdollinen alaotsikko

Vuosi 2022

Kieli suomi

Sivumäärä 43 + 3 liitettä

Ohjaaja Etunimi Sukunimi

Tiivistelmässä käytetään rivinväliä 1. Tiivistelmä kirjoitetaan täydellisinä lauseina. Sähketyyliä ei käytetä, ei liioin alleviivauksia, kursivointeja tai harvennuksia. Tiivistelmän pituus on 75 - 200 sanaa. Teksti on hyvä jakaa kappaleisiin luettavuuden takia, esimerkiksi seuraavan kaavan mukaan:

Ensimmäisessä kappaleessa kuvaillaan lyhyesti tutkimuksen taustaa ja tutkimusongelmaa sekä tutkimuksen tehtävää.

Toisessa kappaleessa kuvaillaan lyhyesti tutkimuksen teoreettista viitekehystä, keskeisiä käsitteitä sekä tutkimuksessa käytettyjä menetelmiä ja tutkimusaineistoa.

Kolmannessa kappaleessa esitetään lyhyesti tutkimuksen keskeisiä havaintoja ja tuloksia sekä keskeisiä johtopäätöksiä ja tulkintoja.

Nouda Fintosta (n.d.) avainsanat.

Avainsanat oppiminen, tehokkuus, opetusmenetelmät  
 (4–5 avainsanaa)

VAASAN AMMATTIKORKEAKOULU

UNIVERSITY OF APPLIED SCIENCES (font size 12)

Name of the Degree Programme (e.g. Sähkötekniikka/ Liiketalous/ Hoitotyö, font size 12)

**ABSTRACT (font size 14)**

Author First name Last name

Title Title of the Thesis

Possible subtitle

Year 2022

Language Finnish

Pages 43 + 3 Appendices

Name of Supervisor First name last name

The abstract is written here using single spacing. The abstract should briefly state the background, aim, material and methods, as well as results and conclusions of the thesis.

Always use the official name of your degree programme (virallinen koulutusohjelman nimi on suomenkielisissä ohjelmissa suomenkielinen).

Keywords learning, stimulation, teaching methods

**SISÄLLYS**

Tiivistelmä

ABSTRACT

[1 Johdanto 6](#_Toc146830388)

[2 Projektin tausta ja tavoitteet 7](#_Toc146830389)

[3 TEORIA 8](#_Toc146830390)

[3.1 Pilvipalvelu 8](#_Toc146830391)

[3.2 Software as service(SaaS) 9](#_Toc146830392)

[3.3 PaaS (Platform as a service) 11](#_Toc146830393)

[3.4 (Instracture as a service) 12](#_Toc146830394)

[3.5 Domainhotelli.fi 13](#_Toc146830395)

[3.6 Digital Ocean 14](#_Toc146830396)

[3.7 Nginx 16](#_Toc146830397)

[4 projektin lähestymistapa ja toteutus 17](#_Toc146830398)

[4.1 Kaaviot 17](#_Toc146830399)

[4.2 Tietoturva 17](#_Toc146830400)

[4.3 Ohjelmsiston testaus 17](#_Toc146830401)

[5 projektin tuotokset 18](#_Toc146830402)

[6 johtopäätökset ja tulkinta 22](#_Toc146830403)

[Lähteet 24](#_Toc146830404)

**KUVIO- JA TAULUKKOLUETTELO**

[**Kuvio 1.** Tyylin valintaikkuna. (MS Office ruutukaappaus) **Virhe. Kirjanmerkkiä ei ole määritetty.**](#_Toc440028603)

[**Taulukko 1.** Opinnäytetyön sisällysluetteloesimerkit. **Virhe. Kirjanmerkkiä ei ole määritetty.**](#_Toc138776544)

# Johdanto

Yleisurheilutapahtumat ovat suosittuja liikuntatapahtumia, jotka tuovat yhteen urheilijoita ja yleisöä ympäri maailmaa. Tällaiset tapahtumat ovat erinomainen tilaisuus paitsi urheilun seuraamiseen myös osallistumiseen. Kalevan Yleisurheilutapahtuma, joka tunnetaan myös nimellä Kalevan kisat, on yksi Suomen merkittävimmistä yleisurheilutapahtumista. Se houkuttelee vuosittain tuhansia urheilijoita ja katsojia.

Tämän opinnäytetyön tavoitteena on kehittää verkkosovellus, joka helpottaa Kalevan Yleisurheilutapahtuman hallintaa ja ilmoittautumista. Tämä sovellus tarjoaa käyttäjille mahdollisuuden ilmoittautua tapahtuman vapaaehtoisiksi ja antaa ylläpitäjille mahdollisuuden hallita ilmoittautuneiden tietoja tehokkaasti. Tämän tyyppinen sovellus voi parantaa tapahtuman organisointia ja osallistujien kokemusta merkittävästi.

Tutkimuskysymys, johon tämä opinnäytetyö pyrkii vastaamaan, on: "Kuinka voimme suunnitella ja kehittää verkkosovelluksen, joka mahdollistaa ilmoittautumisen Kalevan Yleisurheilutapahtumaan ja tarjoaa ylläpitäjille tehokkaat hallintatyökalut?" Tämä tutkimuskysymys nousee tarpeesta parantaa Kalevan Yleisurheilutapahtuman hallintaprosesseja ja tehdä ilmoittautumisesta vaivattomampaa sekä urheilijoille että tapahtuman järjestäjille.

Tulemme tutkimaan ja analysoimaan verkkosovelluksen suunnittelun ja kehityksen vaiheita, käytettyjä menetelmiä sekä saatuja tuloksia myöhemmissä osioissa tässä opinnäytetyössä

# Projektin tausta ja tavoitteet

Yleisurheilutapahtumien hallinta ja koordinointi ovat perinteisesti olleet monimutkaisia ja aikaa vieviä prosesseja, jotka vaativat tarkkuutta ja tehokasta tiedonkäsittelyä. Tämä projekti syntyi tarpeesta parantaa Kalevan yleisurheilutapahtuman järjestämistä ja tehdä siitä saumattomampi kokemus sekä osallistujille että järjestäjille.

Aikaisemmat yleisurheilutapahtumat ovat tuoneet esiin haasteita, kuten ilmoittautumisprosessin monimutkaisuus, tiedonhallinnan hajanaisuus ja tiedonkulun esteet ylläpitäjien ja osallistujien välillä. Nämä haasteet ovat synnyttäneet tarpeen digitaaliselle ratkaisulle, joka tekee ylläpitäjien tehtävistä helpompia ja antaa osallistujille paremman kokemuksen. Tämän sovelluksen keskeiset tavoitteet ovat:

* Ilmoittautuminen ylläpitäjäksi: Sovellus mahdollistaa käyttäjille helpon ilmoittautumisen ylläpitäjiksi tapahtuman hallintaa varten.
* Tietojen hallinta: Ylläpitäjät voivat lisätä, päivittää ja poistaa tietoja Kalevan yleisurheilutapahtuman toimeksiantajien tietoihin liittyen.
* Excel-tiedostojen tuonti ja vienti: Sovellus tarjoaa mahdollisuuden ladata toimeksiantajien tiedot Excel-tiedostona, mikä tekee tiedon käsittelystä helppoa ja tehokasta.

# TEORIA

## Pilvipalvelu

**Pilvipalvelu** tarkoittaa internetin yli tarjottavia tietojenkäsittelypalveluita [1]. Esimerkkejä pilvipalvelun tarjoajia ovat Microsoft, Google ja Amazon, DigitalOcean ja MongoDB. Näistä MongoDB tarjoaa infrakstuurin, DigitalOcean tarjoaa mm. infrastruktuurin ja virtuaalikoneita. Microsoft, Google ja Amazon tarjoavat infrastruktuurin, ohjelmistoalustoja sekä ohjelmistoja.

Yritykset käyttävät pilvipalveluja, koska datan käyttäminen ei ole käyttäjän sijainnista riippuvainen. Käyttäjä voi päästä käsiksi pilvipalveluun tallennettuihin tietoihin millä laitteella tahansa, jolla voi käyttää internettiä. Pilvipalvelut ovat myös yrityksille taloudellisesti kannattavaa, koska yritykset maksavat vain siitä, mitä ne käyttävät, sekä palvelinten ylläpitämiseen tarvittavat laitteistot ovat kalliita. Laitteistoja voi joutua korjauttamaan tai jopa vaihtamaan vikojen vuoksi, mikä aiheuttaa yritykselle kuluja. Jos yritykset eivät ylläpidä omia palvelimia, yritysten ei tarvitse ottaa kantaa palvelimien ylläpitoon ja asentamiseen liittyviin ongelmiin [2].

Pilvipalvelut voidaan jakaa erilaisiin tyyppeihin, esimerkiksi SaaS(Software as service, , IaaS (Infrastructure as a Service) ja PaaS (Platform as Service). Pilvipalveluiden hinnoittelu perustuu yleensä käytettävän datan mukaan.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, muotoilu

Kuvaus luotu automaattisesti

**Kuvio 1**. (https://www.stackscale.com/blog/cloud-service-models/)

## Software as service(SaaS)

Software as a Service on pilvipalvelutyyppi, jossa pilvipalveluntarjoajan ohjelmisto on tarjolla pilvessä. SaaS-tyyppisessä pilvipalvelumallissa palveluntarjoajalla on täysi hallinta infrastruktuuriin, suorituskykyyn, turvallisuuteen, skaalautuvuuten, yksityiseen ja myös paljon muihin ominaisuuksiin Architecting the Cloud, Michael J.Kavis, s.59). SaaS-ratkaisuihin sisältyy monenlaisia kategorioita. Yleisimmät näistä ovat yritysten liiketoimintasovellukset (ERP), kirjanpito, henkilöstö ja palkanlaskenta

Yritykset käyttävät paljon SaaS-tyyppistä pilvipalveluntarjoajaa silloin, kun tarvitaan osaamista, joka ei kuulu ydinosaamiseen (Architecting the Cloud, Michael J.Kavis, s.60). ”

Yritykset päättävät luottaa SaaS-ratkaisuihin ei-ydintoimintoihin, jotta niiden ei tarvitse tukea sovellusinfrastruktuuria, tarjota ylläpitoa ja palkata henkilöstöä hallitsemaan kaikkea” (Architecting the Cloud, Michael J. Kavis, s.60”. SaaS-tyyppinen pilvipalvelu on hyvä valinta esimerkiksi silloin, kun ohjelmistoa käytetään mobiililaitteessa tai web-ympäristössä, halutaan vähentää järjestelykustannuksia, halutaan saada ohjelmisto tuotantoon nopeasti tai kun pilvi-/SOA:sta osaaminen on vähäinen (Architectectin the Cloud, Mickael J.Kavis, s.60). Esimerkiksi SaaS-tyyppisiä pilvipalveluita ovat Google Workspace, Salesforce. SaaS ja PaaS ero on se, että että SaaStyyppinen pilvipalvelu tarjoaa ohjelmistokehitystyökaluja koko ohjelmiston hallinnointiin, esimerkiksi bugien korjaus ja päivityksejuhmjjpokopkt, kun taas Paas tarjoaa sovelluskehitystyökaluja sekä suoritusympäristöjä.

Pilvipalvelutarjoajan asiakkaan ei tarvitse huolehtia myöskään siitä, millä ohjelmointikielellä ohjelmisto tehdään sekä miten ohjelmisto isännöidään, kun käytetään SaaS-tyyppistä pilvipalvelua.

## PaaS (Platform as a service)

Platform as a service on pilvipalvelutyyppi, jossa palvelualusta ulkoistetaan. On olemassa ilmaisia avoimen lähdekoodin ohjelmistoja sekä maksullisia suljetun lähdekoodin ohjelmistoja.

Avoimen lähdekoodin etu suljettuun lähdekoodiin nähden on se, että kuluttajan ei tarvitse miettiä uuden palveluntarjoajan etsimistä, jos sen hetkinen palveluntarjoaja lopettaa tuotteensa myymisen (Architecting the Cloud, Michael J.Kavis, s.63).

PaaS-tyyppiset pilvipalveluntarjoajien ohjelmistot jaetaan julkisiin palveluihin ja yksityisiin palveluihin. Julkinen PaaS-tyyppinen pilvipalveluntarjoaja hallinnoi asiakkaan taustalla olevaa infrastruktuuria, verkkoja, tallennustilaa, laitteita ja käyttöjärjestelmiä. Tietoturvakorjaukset, lokikirjaukset, valvonnan, skaalaus-, vikasieto- ja muut järjestelmänhallintaan liittyvät tehtävät suoritetaan PaaS-tyyppisessä pilvipalvelun palvelussa, jolloin ohjelmistokehittäjien työ nopeutuu ja pilvivalmiiden sovellusten rakentaminen on helppoa (Architecting the Cloud, Michael J.Kavis, s63). Yksityiset PaaS-tyyppiset pilvipalveluntarjoajat eivät tarjoa niin abstraktisia tehtäviä kuin julkiset pilvipalveluntarjoajat. Yksityiset PaaS-tyyppiset pilvipalveluntarjoajat tarjoaa omaa ohjelmistoa käyttöön yksityisessä ja julkisessa pilvessä, mutta samalla ottaa kantaa asiakkaan sovelluspinon ja infrastruktuurin hallintaan.

(Architecting the Cloud, Michael J.Kavis, s.63).

PaaS sisältää tarvittavat sovellukset, joiden avulla voidaan kehittää ohjelmistot kuluttajien käyttöön. Esimerkiksi Dokku ja Heroku ovat PaaS-tyyppisiä pilvipalveluita. Tyypillinen veloitus PaaS-tuotteilla on kuukausittainen. Ero IaaS ja PaaS välillä on se, että IaaS antaa paremman hallinnan käyttöjärjestelmästä, kun taas PaaS antaa kuluttajille enemmän joustavuutta ja helppokäyttöisyyttä. PaaS -tyyppinen pilvipalvelu on hyvä valinta silloin, kun halutaan nopeasti käyttää sovellusta ottamatta kantaa taustalla pyörivään infrastruktuuriin. PaaS -tyyppinen pilvipalvelu taas ei ole hyvä silloin, kun halutaan ohjelmistolle enemmän valvontaa sekä räätälöintiä.

## (Instracture as a service)

Infrastructure as a service on pilvipalvelutyyppi, johon sisältyy vain yksi seuraavista: verkkoyhteydet, tallennustila, palvelimet ja niiden ylläpito. ” Jos sovelluksella tai palvelulla on suorituskyky- tai skaalautuvuusvaatimuksia, jotka edellyttävät kehittäjien hallitsevan muistia, määrittävät tietokantapalvelimet ja sovelluspalvelimet suorituskyvyn maksimoimiseksi, määrittävät, kuinka tiedot jaetaan levyn karaille, manipuloimaan käyttöjärjestelmää ja niin edelleen, sinun tulee hyödyntää IaaS” (Architecting The cCloud, Michael J.Kavis, s.65). IaaS on tällöin taloudellisesti kannattavampaa kuin PaaS, jos datamäärät ovat isoja(Architecting The Cloud, Michael J.Kavis, s65). IaaS kannattaa käyttää PaaS sijaan myös seisokkien riskien vähentämisen takia. Jos PaaS:in pilvipalvelutarjoajalla on ongelmia internet yhteyden kanssa, asiakas joutuu odottamaan, kunnes pääsee käyttämään PaaS-pilvipalvelutarjoajan palveluita. Jos asiakas on valinnut IaaS PaaS:in sijaan, asiakas voi kehittää liiketoimintaansa, jolloin hän voi rakentaa redundantteja palveluita fyysisiin tai virtuaalisiin palvelinkeskuksiin (Architecting The Cloud, Michael J.Kavis, s.66). rakentaa redundantteja palveluita useisiin fyysisiin tai virtuaalisiin palvelinkeskuksiin Datan laitteiden hankinta, ylläpitäminen ja korjaaminen on myös kallista. IaaS-tyyppisessä pilvipalvelussa on myös haittoja. Jos yrityksellä on käytössä tietoturvakriittistä dataa, niin tietoturvaan on panostettava. Yritys joutuu siirtämään tietoturvan vastuun tällöin pilvipalvelutarjoajalle, eikä voi tällöin hallita niin paljoa tietoturvaa. Esimerkiksi DigitalOcean ja MongoDB ovat IaaS-tyyppisiä palvelun tarjoajia.

## Domainhotelli.fi

Domainhotelli on suomalainen yritys, joka tarjoaa nimipalveluja (DNS) eri palveluille sekä Web-hotelleja. ” Jokainen verkkotunnus löytyy nimipalvelinjärjestelmästä eli DNS:stä, jossa verkkotunnus osoittaa yleensä tiettyyn IP-osoitteeseen. Nimipalvelussa määrätään esimerkiksi, minne kotisivut ja sähköpostit ohjautuvat ”(<https://www.domainhotelli.fi/asiakkaat/ohjeet/15/Mika-on-domain.html>). Seuraavassa kuvassa esiintyy esimerkkejä Domainhotellin hinnoittelusta.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, ympyrä, Fontti

Kuvaus luotu automaattisesti

**Kuvio 2.** Domainhotellin hinnoittelumallit. (<https://www.domainhotelli.fi/webhotelli/>)

Tässä ohjelmistossa on käytetty Domainhotellin tarjoamaa 3 tähden web-hotellia. Ohjelmisto käyttää web-hotellin tietokantaa, joka käyttää SQL-ohjelmointikieltä. Ohjelmiston verkkotunnus on ostettu myös Domainhotellista. Verkkotunnus tarkoittaa internet-osoitetta, esimeriksi [www.domainhotelli.fi](http://www.domainhotelli.fi) (https://www.domainhotelli.fi/asiakkaat/ohjeet/15/Mika-on-domain.html).

## Digital Ocean

Digital Ocean on yhdysvaltalainen pilvipalveluntarjoaja, joka tarjoaa erilaisia pilvipalveluita kehittäjille, startup-yrityksille ja IT-ammattilaisille. Yhtiö perustettiin vuonna 2011, ja se on kasvanut suosituksi vaihtoehdoksi pilvipalvelualustojen, kuten Amazon Web Services ja Microsoft Azure rinnalla (https://www.digitalocean.com/).

Digital Ocean tarjoaa muun muassa seuraavia pilvipalveluja: dropletit, kuva- ja tallennuspalvelut, Digtal Ocean Kubernetes, tietokantapalvelut (<https://www.digitalocean.com/>). Digital Ocean Kubernetesissa käytetään Kubernetes-ohjelmistoa (https://docs.digitalocean.com/products/kubernetes/), joka on avoimen lähdekoodin järjestelmä, jota käytetään konttisovellusten käyttöönotossa, skaalauksen ja hallinnan automatisoinissa (<https://kubernetes.io/>). Digital Ocean Kubernetes mahdollistaa sen, että palveluntarjoajan käyttäjän ei tarvitse ottaa kantaa konttiinfrastruktuurin käsittelyyn (<https://docs.digitalocean.com/products/kubernetes/>).

Digital Oceanin tarjoamat tietokantapalveluissa käytetään MongoDB, PostgeSQL , MySQL ja Redis (https://www.digitalocean.com/products/managed-databases). Digital Oceanin tietokantapalvelujen hinnoittelu perustuu varatun datamäärän suuruuden mukaan. Mitä isompi on varattu tallennustila, sitä enemmän se maksaa. Veloitus on kuukausittainen.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, numero, Fontti

Kuvaus luotu automaattisesti

**Kuvio 3**. Digital Oceanin ylläpitämän MySQL-tietokannan hinnoittelumalli

Tässä ohjelmistossa on käytetty Digital Oceanin tarjoamaa Droplettia. ” Digital Ocean Droplets ovat Linux-pohjaisia ​​virtuaalikoneita (VM), jotka toimivat virtualisoidun laitteiston päällä. Jokainen luomasi Droplet on uusi palvelin, jota voit käyttää joko itsenäisenä tai osana suurempaa pilvipohjaista infrastruktuuria” (<https://www.digitalocean.com/>).

Digital Oceanin hinnoittelu perustuu virtuaalikoneen käytetyn muistin mukaan. Mitä enemmän muistia virtuaalikoneella on käytettävissä, sitä kalliimpi käytettävä virtuaalikone on. Seuraavassa kuvassa Digital Oceanin virtuaalikoneen hinnoittelumalli.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, numero, Fontti

Kuvaus luotu automaattisesti

**Kuvio 4**. Digital Oceanin virtuaalikoneen hinnoittelumalli (https://www.digitalocean.com/pricing/droplets).

## Nginx

Nginx on avoimen lähdekoodin, suorituskykyinen, monikäyttöinen HTTP- ja käänteisproksi-palvelin, jota käytetään yleisesti verkkosivustojen ja web-sovellusten palvelimina. Se on suunniteltu nopeaksi, tehokkaaksi ja skaalautuvaksi, ja se pystyy käsittelemään suurta määrää samanaikaisia pyyntöjä. Nginx voi toimia staattisten tiedostojen palvelimena, kuorman tasapainottajana tai käänteisproksina, ja se on suosittu ratkaisu, kun tarvitaan korkeaa suorituskykyä ja luotettavuutta web-palvelinten hallinnassa (”<https://nginx.org/>”).

# projektin lähestymistapa ja toteutus

Tässä kappaleessa kerromme, miten projektia lähestyttiin ja toteutettiin. Projekti alkoi suunnitteluvaiheella, jossa määriteltiin tarkat kriteerit ohjelmistolle. Ohjelmiston suunnitteluvaiheessa käytettiin käyttäjätapausdiagrammia, sekvenssikaaviota ja ERD-kaaviota. ERD-diagrammi tarjoaa graafisen esityksen tietokantataulukoista (https://www.visual-paradigm.com/features/database-design-with-erd-tools/), käyttäjätapausdiagrammi taas vastaa kysymykseen ”mitä tehdään”, eikä niinkään kerrot sen tarkempia kriteerejä ohjelmistolle (<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>) ja sekvenssikaavio kertoo missä järjestyksessä tapahtumat etenevät, kun ohjelmistoa käytetään (https://www.visual-paradigm.com/learning/handbooks/software-design-handbook/sequence-diagram.jsp). Kaaviot ovat toteutettu Visual Paradiagram-ohjelmistoa käyttäen. ”Visual ParaDiagram on johtava ja maailmanlaajuisesti tunnettu Business and IT Transformation -ohjelmistoratkaisujen toimittaja. Sen avulla organisaatiot voivat parantaa liiketoiminta- ja IT-ketteryyttä ja edistää innovaatioita suosittujen avoimien standardien avulla ” (<https://www.visual-paradigm.com/aboutus/>).

Suunnitteluvaiheen jälkeen aloitettiin ohjelmointivaihe. Aina kun ohjelmoitiin jotain toiminnallisuutta, kirjoitettiin yksikkötesti JEST-kehystä käyttäen. Tällä tavalla varmistetaan, että ohjelma toimii niin kuin pitääkin.

Projekti on ohjelmoitu Typescript- ja Javascript-ohjelmointikieliä käyttäen. Käyttöliittymä on ohjelmoitu React-kehystä hyödyntäen ja REST-Api toteutettu Node.js kehystä käyttäen. ”REST API on tapa käyttää verkkopalveluja yksinkertaisella ja joustavalla tavalla ilman käsittelyä” (<https://www.geeksforgeeks.org/rest-api-introduction/>).

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisesti

**Kuvio 5**. Esimerkki Node.js kehyksellä ja Express-kirjastolla tehty REST-Api.

Tietokanta on toteutettu MySQL-ohjelmointikielellä.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisesti

**Kuvio 6**. Esimerkki taulun luomisesta MySQL-ohjelmointikielellä. (<https://www.w3schools.com/sql/sql_create_table.asp>)

## Kaaviot

Kuva, joka sisältää kohteen teksti, kuvakaappaus

Kuvaus luotu automaattisesti

**Kuvio 7**.

Kuva, joka sisältää kohteen teksti, kuvakaappaus

Kuvaus luotu automaattisesti

**Kuvio 8**. ERD-diagrammi

Kuva, joka sisältää kohteen teksti, diagrammi, kuvakaappaus, viiva

Kuvaus luotu automaattisesti

**Kuvio 9**. Käyttäjätapauskaavio.

## Tietoturva

Tässä kappaleessa käydään läpi, miten tietoturva on otettu huomioon tämän opinnäytetyön sovelluksessa.

SQL-injektio on menetelmä, jonka avulla voidaan hyödyntää SQL-ohjelmointikielellä tehdyn tietokannan haavoittuvuuksia. (https://www.geeksforgeeks.org/sql-injection/). Käytännössä tämä toimii siten, että jos selaimen mahdolliset sisältämät tekstikentät sisältävät SQL-koodia ja tekstikentän sisältö lähetetään palvelimelle, haitallinen SQL-koodi pystytään tällöin suorittamaan, mikäli tekstikentän sisältöä ei ole erikseen tarkistettu.

Kuva, joka sisältää kohteen teksti, luonnos, diagrammi, Ihmisen kasvot

Kuvaus luotu automaattisesti

**Kuvio 10**. SQL-injektio. (<https://www.geeksforgeeks.org/sql-injection/)>.)

SQL-injektion suojautumiseksi ohjelmistossa käytetään SQL komennoissa parametrisoituja SQL-komentoja. Kun käytetään parametrisoituja SQL-komentoja, SQL-koodi tarkistetaan ennen sen suorittamista. Jos komento sisältää haitallista SQL-koodia, jolla pystytään manipuloimaan tietokantaa, komento keskeytetään. Seuraavassa kuvassa näytetään esimerkki käyttäjän lisäämisestä tietokantaan, jossa käytetään parametrisoitua SQL-komentoa.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, ohjelmisto

Kuvaus luotu automaattisesti

**Kuvio 11**. Käyttäjien lähettäminen tietokantaan

Domainhotellin ylläpitämässä tietokannassa on ominaisuus, jolla pystytään määrittämään käyttäjille oikeudet, miten he voivat muokata tietokantaa, esimerkiksi voivatko poistaa dataa, muokata, lisätä jne. Ohjelmisto käyttää ainoastaan INSERT, UPDATE, DELETE ja SELECT komentoja, joten on hyvä, että esim. DROP-komentoa ei sallita suoritettavan, koska DROP-ominaisuudelle ei ole millään käyttäjällä tarvetta. DROP-komento poistaa tietokannasta kokonaisen taulun.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, numero, ohjelmisto

Kuvaus luotu automaattisesti

**Kuvio 12**. Domainhotellit käyttäjäoikeudet

Tietokannan käyttäjänimet ja salasanat kannattaa kryptata. Kryptaus tarkottaa menetelmää, jossa merkkijono muutetaan toiseen muotoon matemaattisia kaavoja käyttäen (<https://www.geeksforgeeks.org/introduction-to-hashing-data-structure-and-algorithm-tutorials/>). Kryptaus tehdään siksi, koska jos tietokannan tietoihin pääsee käsiksi, tiedot ovat vaikeasti tulkittavissa. Kun käyttäjien tietoja ei voida mitenkään tulkita, kukaan, jolle ei ole myönnetty Admin-tunnuksia, ei voi päästä Admin-oikeuksiin käsiksi. Tässä ohjelmistossa käytetään SHA-265 algoritmia. Esimerkiksi sana ”Hello World!” on SHA-256 algoritmia käyttäen kryptatuna”7f83b1657ff1fc53b92dc18148a1d65dfc2d4b1fa3d677284addd200126d9069”

Kuva, joka sisältää kohteen teksti, kuvakaappaus, ohjelmisto, Multimediaohjelmisto

Kuvaus luotu automaattisesti**Kuvio 13**. Admin käyttäjän todentaminen.

Kuvassa (muunna kuvan numero) nähdään, kuinka admin käyttäjä todennetaan. Selaimella kysytään Admin-käyttäjän käyttäjänimeä ja salasanaa. Koska käyttäjänimi ja salasana ovat kryptattuja, joudutaan myös lomakkeen kyseltyjen arvot kryptaamaan ja vertaamaan niitä tietokannan sisältöön. Mikäli käyttäjä löytyy, jatketaan ohjelman suorittamista.

Autentikaatio tarkoittaa menetelmää, jonka avulla pystytään todentaa käyttäjä, jolla on oikeus käyttää sille tarkoitettuja verkkoresursseja. (<https://www.geeksforgeeks.org/sql-injection/)>. Tässä sovelluksessa käytetään kahta autentikaatiomenetelmää, jotka ovat JSON Web token(JWT) ja monivaiheinen tunnistautuminen(MFA).

”JSON-verkkotunnus (JWT) on JSON-objekti, jota käytetään tiedon siirtämiseen turvallisesti verkon yli (kahden osapuolen välillä). Sitä voidaan käyttää todennusjärjestelmässä ja sitä voidaan käyttää myös tiedonvaihtoon. Tunnus koostuu pääasiassa otsikosta, hyötykuormasta ja allekirjoituksesta.” (<https://www.geeksforgeeks.org/sql-injection/)>.

” Monitekijätodennus (MFA) käyttää kahta tai useampaa todennusmenetelmää eri luokista käyttäjän henkilöllisyyden vahvistamiseksi. MFA on yhä tärkeämpi suojatuissa verkoissa. Se on kaksivaiheinen vahvistusmekanismi, joka tyydyttää käyttäjien vaatimuksen helposta kirjautumisprosessista ja suojaa samalla tietoja ja sovelluksia. Useiden vahvistusmenetelmien, kuten puhelimen, tekstiviestien ja mobiilisovellusten vahvistuksen, avulla se tarjoaa vankan todennuksen. MFA:n turvallisuus tulee sen kerroksellisesta lähestymistavasta.” (https://www.geeksforgeeks.org/multifactor-authentication/). Tässä opinnäytetyön ohjelmistossa käytetään sähköpostia ja salasanaa tunnistamaan käyttäjän. Kun käyttäjä syöttää ohjelmiston käyttäjätilin sähköpostiosoitteen ja salasanan, ohjelmisto lähettää käyttäjän sähköpostiin tunnistautumiskoodin. Selaimessa kysytään tunnistautumiskoodia. Kun käyttäjä syöttää oikean tunnistautumiskoodin, ohjelmisto tunnistaa käyttäjän ja käyttäjä saa hänelle tarkoitetut oikeudet ohjelmiston käytössä.

Kuva, joka sisältää kohteen vihreä, diagrammi, muotoilu

Kuvaus luotu automaattisesti

**Kuva 2**. (<https://www.geeksforgeeks.org/multifactor-authentication/>)

General Data Protection Regulation (GDPR) on Euroopassa toimiva henkilötietojen käsittelyyn liittyvä laki, joka otettiin ensimmäistä kertaa käyttöön vuonna 2018 (<https://tietosuoja.fi/gdpr>). ” Tietosuoja-asetus antaa paremman suojan henkilötiedoillesi ja enemmän keinoja hallita tietojesi käsittelyä” (<https://tietosuoja.fi/gdpr>). Orgaanisaatio voi GDPR mukaan kerätä tietoja, jos sillä on hyvä peruste (<https://tietosuoja.fi/gdpr>). Tässä on esimerkkejä hyviä perusteista: rekiströidyn suostumus, sopimus, rekisterinpitäjän lakisääteinen velvoite, elintärkeiden etujen suojaaminen, yleinen etu ja julkinen valta, rekisterinpitäjän tai kolmannen osapuolen oikeutettu etu. Tietojen suojaamista varten GDPR mukaan tietojen keräävän organisaation on nimitettävä tietosuojavastaava, mikäli joku näistä seuraavista täyttyy: 1) Organisaatio käsittelee laajamittaisesti arkaluonteisia tietoja, 2) Seuraa ihmisiä laajamittaisesti 3) Organisaatio on julkishallinnon toimija (<https://tietosuoja.fi/gdpr>). GDPR mukaan organisaation on kerrottava seuraavat tiedot:

* organisaatiosi nimi, yhteystiedot ja mahdollisen tietosuojavastaavan yhteystiedot
* mitä tietoja organisaatiosi aikoo käsitellä
* mihin tarkoitukseen organisaatiosi käsittelee henkilötietoja
* henkilötietojen käsittelyn oikeusperuste
* kuinka kauan tietoja säilytetään
* mikä toinen organisaatio tai henkilö voi saada tiedot
* siirretäänkö henkilötietoja EU:n ulkopuolelle
* henkilön tietosuojaoikeudet
* oikeus tehdä valitus tietosuojaviranomaiselle
* oikeus milloin tahansa peruuttaa suostumus, jos käsittely perustuu suostumukseen
* tiedot automaattisesta päätöksenteosta ja käsittelyyn liittyvästä logiikasta.

Nämä kriteerit ohjelmistossa on täytetty. Kalevan kisojen yhteystiedot näkyvät selkeästi rekisteröintilomakkeen alla sekä on kerrottu, että tiedot poistetaan kolme päivää kisojen päättymisen jälkeen. Tietoja käsitellään vain Kalevan kisojen järjestäjien kesken, eikä tietoja levitetä muille.

SSL (Secure Sockets Layer) ja sen seuraaja TLS (Transport Layer Security) ovat protokollia, joita käytetään luomaan todennettuja ja salattuja linkkejä verkkoon kytkettyjen tietokoneiden välille. Vaikka SSL-protokolla vanhentui TLS 1.0:n julkaisun myötä vuonna 1999, näihin liittyviin teknologioihin viitataan edelleen nimellä "SSL" tai "SSL/TLS". Uusin versio on TLS 1.3, joka on määritelty RFC 8446:ssa (elokuu 2018). SSL-sertifikaatti sisältää myös tunnistetiedot verkkosivustosta, mukaan lukien sen verkkotunnus ja valinnaisesti tunnistetiedot sivuston omistajasta. Jos verkkopalvelimen SSL-varmenteen on allekirjoittanut julkisesti luotettu sertifikaattiviranomainen (CA), kuten SSL.com, loppukäyttäjien verkkoselaimet ja käyttöjärjestelmät luottavat palvelimelta digitaalisesti allekirjoitettuihin sisältöihin aitoina. // Muuta tämä SSL omin sanoin. Tietoturvan kannalta SSL-sertifikaatti on olennainen web-ohjelmistossa. SSL-sertifikaatti mahdollistaa sen, että hakkerit eivät pääse varastamaan tai näkemään siirrettyjä tietoja.

How SSL certificates work


**Kuvio 14**. [www.hostinger.com](http://www.hostinger.com) SSL-Toiminta

”Palvelunestohyökkäys (DoS) keskittyy siihen, että resurssi (sivusto, sovellus, palvelin) ei ole käytettävissä siihen tarkoitukseen, joka se on suunniteltu. On monia tapoja tehdä palvelusta laillisten käyttäjien käytettävissä muun muassa manipuloimalla verkkopaketteja, ohjelmointia, loogisia tai resurssien käsittelyn haavoittuvuuksia. Jos palvelu saa erittäin suuren määrän pyyntöjä, se voi lakata olemasta laillisten käyttäjien saatavilla. Samalla tavalla palvelu voi pysähtyä, jos ohjelmahaavoittuvuutta hyödynnetään tai tapaa, jolla palvelu käsittelee käyttämiään resursseja” (<https://owasp.org/www-community/attacks/Denial_of_Service>). Tässä ohjelmistossa on suojauduttu palvelunestohyökkäyksellä siten, että tietyltä laitteelta on sallittu kymmenen yhteyspyyntöä per sekunti. Mikäli sallittu yhteyspyyntö menee yli, yhteyspyyntöä ei käsitellä ja ohjelmisto toimii siitä huolimatta.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisesti

**Kuvio 15**. Nginx konfiguraatio. DDOS-suojaus

Ohjelmistossa koodirivi ” limit\_req\_zone $binary\_remote\_addr zone=mylimit:10m rate=10r/s;” ja locaton-lohkossa määritelty limit\_req zone=mylimit määrittävät, että jokaista tiettyä ip-osoitetta kohden saa olla vain ja ainoastaan kymmenen yhteyspyyntöä per sekunti.

## Ohjelmsiston testaus

Ohjelmiston testaus on toteutettu yksikkötesteillä sekä Beta-testillä. ” Yksikkötestaus on eräänlainen ohjelmistotestaus, joka keskittyy ohjelmistojärjestelmän yksittäisiin yksiköihin tai komponentteihin. Yksikkötestauksen tarkoituksena on varmistaa, että ohjelmiston jokainen yksikkö toimii tarkoitetulla tavalla ja täyttää vaatimukset” (<https://www.geeksforgeeks.org/unit-testing-software-testing/>).

Beta-testing on testausmenetelmä, jossa käyttäjät testaavat ohjelmiston toimivuutta. ” Betatestaus auttaa minimoimaan tuotteen epäonnistumisriskit ja parantaa tuotteen laatua asiakkaan validoinnin kautta.” (<https://www.geeksforgeeks.org/unit-testing-software-testing/>). Tässä Beta-testauksen testaajana on ohjelmiston tilaaja, eli Vaasan Kalevan kisan yksi järjestäjistä. Ohjelmiston toiminnallisuuden testaaminen sujui hyvin ja ohjelmisto toimii niin kuin pitääkin. Kaikki toiminnot on testattu, eli Excelin lataaminen, käyttäjien lisääminen tietokantaan, käyttäjien tietojen poistaminen sekä käyttäjän tietojen päivittäminen.

Ohjelmistoon sisältyy neljä yksikkötestitapausta, jossa testataan seuraavat asiat 1) Admin voi kirjautua sisään ja voi nähdä tietokannan tietoja 2) Admin voi poistaaa tietokannan tietoja 3) Admin voi päivittää tietokannan tietoja 4) Käyttäjä voi lähettää tietonsa palvelimelle. Ohjelmiston yksikkötestaukset on toteutettu JEST-kehystä käyttäen.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, näyttö

Kuvaus luotu automaattisesti

**Kuvio 16.** Esimerkki testitapauksesta, jossa käyttäjä voi lähettää tietonsa palvelimelle.

# projektin tuotokset

Tämä opinnäytetyö on tuottanut useita merkittäviä projektituotoksia, jotka ovat olleet olennainen osa Kalevan yleisurheilutapahtuman web-ohjelmiston kehitysprosessia. Seuraavassa esittelemme nämä tuotokset yksityiskohtaisesti:

Keskeisin tuotos tässä projektissa on Kalevan yleisurheilutapahtuman web-ohjelmisto. Tämä ohjelmisto mahdollistaa käyttäjille ilmoittautumisen ylläpitäjiksi ja tarjoaa administraattoreille tietokannan hallintatyökalut. Ohjelmisto on suunniteltu ja toteutettu käyttäjäystävälliseksi ja turvalliseksi, ja se toimii tärkeänä välineenä tapahtuman järjestämisessä ja tiedonhallinnassa.

Web-ohjelmiston keskeisin ominaisuus on käyttäjärekisteri ja ilmoittautumisjärjestelmä. Käyttäjät voivat rekisteröityä tapahtuman ylläpitäjiksi ja ilmoittautua tapahtumaan tämän järjestelmän avulla. Järjestelmä helpottaa ilmoittautumisten seuraamista ja hallintaa administraattoreille.

Projektissa luotu tietokanta on toinen merkittävä tuotos. Se tallentaa käyttäjien ja tapahtuman liittyvät tiedot tehokkaasti ja turvallisesti. Tietoturvassa on otettu huomioon SQL-injektiolta suojautuminen sekä tietokannan kriittiset tiedot ovat muutettu SHA-256 algoritmilla. Administraattorit voivat käyttää tietokantaa tietojen hallintaan ja seurantaan, mikä tehostaa tapahtuman järjestelyjä.

Projektin tulokseen kuuluvat myös kattavat käyttöohjeet ja dokumentaatio. Nämä asiakirjat auttavat sekä käyttäjiä että tulevia kehittäjiä ymmärtämään web-ohjelmiston toimintaa ja käyttöä. Käyttöohjeet sisältävät vaiheittaiset ohjeet rekisteröitymiseen ja ilmoittautumiseen, kun taas dokumentaatio tarjoaa yksityiskohtaisen kuvauksen ohjelmiston arkkitehtuurista ja teknisestä toteutuksesta.

Projektin aikana esiin nousseet kehitysehdotukset ja tulevaisuuden laajennukset ovat myös osa tuloksia. Näitä ehdotuksia voidaan käyttää pohjana ohjelmiston jatkokehitykselle ja parannuksille. Ne voivat sisältää esimerkiksi reaaliaikaisen tiedonseurannan, tilastolliset raportit ja mobiilisovelluksen kehittämisen.

Lisäksi projektin saavutukset ja oppimiskokemus ovat tärkeä osa tuloksia. Projekti on tarjonnut mahdollisuuden hankkia arvokasta käytännön kokemusta web-ohjelmiston kehityksestä ja projektinhallinnasta. Projekti myös opetti datan kulujen kustomoinnissa. Palveluntarjoajalla on väliä, kun etsitään palveluja. Hinnoilla on iso merkitys ja eri palveluntarjoajien tietyt tuotteet sopivat erilaisiin tilanteisiin. Esimerkiksi tässä projektissa alustavasti mietittiin Amazon EC2-virtuaalikonetta. Päädyin lopuksi käyttämään Digital Oceanin virtuaalikoneita, koska niillä oli kiinteä hinnoittelu tietylle resurssille. Seuraavassa kuvassa nähdään Amazonin virtuaalikoneiden hinnoittelumalli.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, numero, dokumentti

Kuvaus luotu automaattisesti

**Kuva 17.** Amazon EC2hinnoittelumalli(https://aws.amazon.com/ec2/pricing/on-demand/)

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisesti

**Kuva 18**. Amazon virtuaalikoneen datasiirron hinnoittelumalli”https://aws.amazon.com/ec2/pricing/on-demand/”

Huomataan, että Amazonin hinnoittelu perustuu myös siihen, kuinka paljon dataa virtuaalikone siirtää virtuaalikoneen ulkopuolelle. Digital Oceanin virtuaalikone kustantaa yhtä paljon Amazonin virtuaalikoneeseen nähden, mutta Amazonin virtuaalikone tulee kalliimmaksi, kun joudutaan siirtämään dataa ulos virtuaalikoneesta esim. tekemään SQL-komentoja tietokantaan, joka ei ole virtuaalikoneen sisällä. Opinnäytetyön Digital Oceanin virtuaalikoneeseen kuuluu 1000 GB verran datan siirtoa virtuaalikoneen ulkopuolelle ja sisälle, joten Digital Oceanin virtuaalikule tulee halvemmaksi. (Kuva x ja kuva x),

Näitä taitoja voidaan hyödyntää tulevissa projekteissa ja ammatillisessa uralla. Yhteenvetona voidaan todeta, että Kalevan yleisurheilutapahtuman web-ohjelmiston kehitysprojekti on tuottanut monipuolisia ja arvokkaita projektituotoksia, jotka edistävät tapahtuman tehokasta järjestelyä ja tietojenhallintaa. Nämä tulokset muodostavat vahvan perustan tuleville parannuksille ja laajennuksille.

# johtopäätökset ja tulkinta

Tässä opinnäytetyössä keskityimme Kalevan yleisurheilutapahtuman tarpeiden täyttämiseen kehittämällä web-ohjelmisto, joka mahdollistaa käyttäjien ilmoittautumisen ylläpitäjiksi ja antaa administraattoreille mahdollisuuden seurata heidän tietojaan tietokannassa. Projektin aikana olemme käyttäneet laajalti erilaisia tekniikoita ja menetelmiä saavuttaaksemme tavoitteemme. Tässä osiossa esitämme johtopäätökset ja tulkinnan opinnäytetyömme tuloksista.

Web-ohjelmiston kehitysprojekti on saatu päätökseen onnistuneesti. Ohjelmisto tarjoaa rekisteröintimahdollisuuden käyttäjille, ja administraattorit voivat helposti seurata ja hallinnoida tietoja tietokannassa. Projektin aikana toteutettujen toimintojen avulla voidaan helpottaa Kalevan yleisurheilutapahtuman järjestelyjä ja tietojenhallintaa.

Käyttäjäkokemus on ollut keskeisessä roolissa web-ohjelmiston suunnittelussa ja toteutuksessa. Käytettävyyden optimointi on parantanut ohjelmiston helppokäyttöisyyttä ja tehnyt siitä houkuttelevan käyttäjille. Käyttäjäpalautteen perusteella voimme todeta, että käyttäjät ovat tyytyväisiä ohjelmiston toimivuuteen ja käyttökokemukseen.

Tietoturva ja tietosuoja ovat ensiarvoisen tärkeitä tekijöitä web-ohjelmiston kehityksessä. Olemme huolellisesti toteuttaneet tietoturvamekanismeja ja tietosuojakäytäntöjä varmistaaksemme käyttäjätietojen turvallisuuden. Tietokantaan tallennetut tiedot on suojattu asianmukaisesti ja käsitelty GDPR-vaatimusten mukaisesti.

Vaikka web-ohjelmisto on saavuttanut tavoitteensa, on aina tilaa parannuksille ja kehitykselle. Tulevaisuuden näkymät sisältävät mahdollisuuden lisätä ominaisuuksia, kuten reaaliaikainen tiedonseuranta, tilastolliset raportit ja käyttäjäystävällisemmän mobiiliversion kehittämisen. Tämä voi edistää entisestään Kalevan yleisurheilutapahtuman tehokkuutta ja osallistujien tyytyväisyyttä.

Tämä opinnäytetyö ei ainoastaan tuonut esille web-ohjelmiston kehittämisen teknisiä taitoja vaan myös tarjosi mahdollisuuden soveltaa näitä taitoja käytännössä todellisessa ympäristössä. Opinnäytetyö on antanut arvokasta tietoa siitä, kuinka teknologian avulla voidaan parantaa ja helpottaa suurten tapahtumien järjestämistä ja tietojenhallintaa.

Kalevan yleisurheilutapahtuman web-ohjelmiston kehitysprojekti on ollut menestys, ja se täyttää tehokkaasti käyttäjien tarpeet. Projektin myötä olemme oppineet arvokkaita taitoja ja saaneet kokemusta ohjelmistokehityksen eri vaiheista. Tulevaisuudessa on mahdollisuus laajentaa ja parantaa ohjelmiston toiminnallisuuksia entisestään, edistäen Kalevan yleisurheilutapahtuman sujuvaa järjestämistä ja osallistujien tyytyväisyyttä.

# Lähteet

Finto. (n.d.). YSO - Yleinen suomalainen ontologia. Noudettu 27.6.2023 osoitteesta <https://finto.fi/yso/fi/>

<https://www.geeksforgeeks.org/rest-api-introduction/>

<https://owasp.org/www-community/attacks/Denial_of_Service>

<https://www.geeksforgeeks.org/multifactor-authentication/>

[www.hostinger.com](http://www.hostinger.com)

<https://tietosuoja.fi/gdpr>

<https://www.geeksforgeeks.org/sql-injection/>

<https://www.digitalocean.com/pricing/droplets>

(<https://www.geeksforgeeks.org/introduction-to-hashing-data-structure-and-algorithm-tutorials/>)

<https://www.w3schools.com/sql/sql_create_table.asp>

<https://nginx.org/>

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

(<https://www.digitalocean.com/>

Architecting the Cloud, Michael J.Kavis

(<https://www.domainhotelli.fi/webhotelli/>

<https://kubernetes.io/>