Full Stack-ohjelmointi

Harjoitustyö

Ari-Matti Rintala, TTC2080-3024

Harjoitustyö

Full Stack – ohjelmointi, Ari Rantala

5.11.2023

Tieto- ja viestintätekniikka

**Sisältö**

[1 Harjoitustyön aihe 2](#_Toc150012469)

[2 Sovelluksen sisältö 3](#_Toc150012470)

[2.1 Sovellusnäkymä ja -toiminnot 3](#_Toc150012471)

[2.2 Tietokannan rakenne 4](#_Toc150012472)

[2.3 Palvelinympäristö 4](#_Toc150012473)

[3 Ohjelmalogiikka 4](#_Toc150012474)

[4 Yhteenveto 7](#_Toc150012475)

[Lähteet 8](#_Toc150012476)

**Kuviot**

[Kuvio 1. Selainnäkymä 3](#_Toc150006976)

[Kuvio 2. Ohjelmalogiikka 5](#_Toc150006977)

# Harjoitustyön aihe

Harjoitustyön tarkoituksena oli rakentaa opintojakson aihealueeseen liittyvä Full Stack -sovellus. Työn aihe oli opiskelijalle vapaavalintainen. Valitsin oman harjoitustyöni aiheeksi juoksuharjotteiden kirjaamiseen ja tilastoimiseen käytettävän sovelluksen toteuttamisen. Valinta tuki mielestäni kurssisisällön vaatimuksia mm. siitä syystä, että sovelluksessa olisi luontevaa käyttää sekä tiedon syöttämistä, että tiedon esittämistä erilaisissa muodoissa. Sovelluksessa myös voitaisiin käsitellä tietokannasta haettua dataa hyvin monenlaisella tavalla.

# Sovelluksen sisältö

## Sovellusnäkymä ja -toiminnot

Jo hyvin varhaisessa vaiheessa sain mielessäni jäsenneltyä asiat, joita haluaisin sovelluksessa toteuttaa. Halusin, että sovelluksen selainnäkymän kautta syötettäisiin uuden harjoituksen tiedot, selaimelle listattaisiin tallennettuja harjoituksia sekä tehtäisiin jonkinlaisia laskutoimituksia tallennettujen harjoitteiden pohjalta. Myöhemmin lisäsin vielä toiminnallisuuden, jolla em. laskutoimituksia ja harjoituksien listauksia tehtäisiin käyttäjän valitsemalta aikajaksolta. Sovellukseen voisi toteuttaa hyvin monenlaisia asioita esimerkiksi harjoitteiden esittämiseen tai ulkoasuun liittyen, mutta nämä eivät toisi itse ohjelmaan teknisellä tasolla lisää kompleksisuutta ainakaan kurssisisällön asioihin liittyen.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, pilvi, muotoilu

Kuvaus luotu automaattisesti

Kuvio . Selainnäkymä

Kuviossa 1 on näkyvillä sovelluksen selainnäkymä. Näkymässä on Uusi Harjoitus -otsikolla oleva ikkuna, joka on ainoa syötteitä vastaanottava osio sivulla. Tähän käyttäjä voi syöttää uuden tallennettavan harjoituksen tiedot. Lisäksi kyseisen osion alalaidassa löytyy valintaosio, jossa näytöllä näytettävät ja tilastoinnissa mukana olevien harjoitteiden aikajakso voidaan valita.

Oikeanpuoleinen ikkuna näyttää valitulta aikaväliltä listattuna harjoitukset ja niiden perustiedot. Samojen harjoitteiden osalta lasketaan sivuston vasemman alalaidan ikkunaan keskiarvoja matkan, harjoitusajan ja keskisykkeen osalta.

## Tietokannan rakenne

Tietokannan tietueiden sisältö koostuu syötettyjen harjoitteiden ajankohdasta päivämäärätarkkuudella (vuosi, kuukausi, päivä), harjoitusmatkasta kilometreinä, harjoitukseen käytetystä ajasta minuutteina ja harjoituksen aikaisesta keskisykkeestä. Näistä ainoa pakollinen tieto harjoitusta syötettäessä on päivämäärä. Tietokantana on käytetty MongoDB-tietokantaa, joka on NoSQL-tyyppinen tietokanta. (Alexander S. Gillis, 2023)

## Palvelinympäristö

Harjoitustyön palvelinympäristö on toteutettu Node.js palvelintekniikalla. Node.js on avoimeen lähdekoodiin perustuva Javascript-ajoympäristö. Node.js on tapahtumakäsittelijä, joka välittää tietoa asyncronisesti palvelimella sen sijaan, että palvelinpyyntöjä toteutettaisiin käyttäjän selaimessa. (Introduction to Node.js, 2023).

# Ohjelmalogiikka

Selainpuolen ohjelma koostuu HTML-koodilla toteutetusta sivustosta, josta on viittaukset tyylitiedostoon sekä Javascript-koodiin, jossa tapahtuu varsinainen tietojen käsittely ja kommunikaatio node.js palvelimelle. Sovelluksen keskiössä on käyttäjän klikattavissa olevat ’tallenna harjoitus’ sekä ajanjakson valintapainikkeet. HTML-sivun auetessa kutsutaan välittömästi Javascript-tiedoston loadLog-funktiota, joka hakee ruudulle tietokantaan tallennetut harjoitukset. Samoin uutta harjoitusta tallennettaessa tai tilastoinnin aikaikkunaa muutettaessa kutsutaan listan päivittävää funktiota, jotta näytetyt harjoitukset ovat aina ajan tasalla.

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

Kuvaus luotu automaattisesti

Kuvio . Ohjelmalogiikka

Kuviossa 2 on esitetty graafisella tasolla ohjelmalogiikka. Sovelluksen keskiössä on web-sivun skriptiosiossa toimivat funktiot (vihreät laatikot). Erityisesti huomioitava on ’clearAndLoad’-funktio, jota käytännössä kutsutaan joka kerta, kun käyttäjä tekee muutoksia sivustolla. Tämä käynnistää ketjureaktion, jossa kutsutaan funktiota toisensa jälkeen siten, että sivustolla näkyvä harjoituslistaus päivittyy ja sitä myöten päivittyy myös tilastointi. Näiden funktioiden sisällä tehdään tarvittava kommunikaatio palvelinpään ja tätä kautta tietokannan kanssa.

Funktioiden toiminta lyhyesti:

* addHarjoitus()
  + Muodostetaan käyttäjän täyttämistä tekstikentistä uusi tietue ennalta määritellyn datarakenteen mukaisesti. Data haetaan input-tekstikentistä ja syötetään Javascript avain-arvo -parirakenteeseen.
  + Syötetään uusi tieto tietokantaan palvelinpyynnöllä.
  + Kutsutaan clearAndLoad-funktiota.
* clearAndLoad()
  + Tyhjätään nykyinen listaus selainnäkymästä, jotta vältytään tuplalistaukselta.
  + Kutsutaan loadLog().
* loadLog()
  + Pyydetään tietokantasisältö palvelimelta.
  + Haetaan käyttäjän tekemä aikajakson valinta selaimelta ja muodostetaan tämän perusteella vertailuajankohta.
  + Tehdään vertailu tietokannan yksittäisten harjoitusten ajankohdan ja em. tilastoajan valinnan välillä.
  + Lisätään aikaikkunaan sopivat harjoitukset uusina HTML-elementteinä selainikkunalle (käyttäen tähän erillistä funktiota). Lisätään kyseiset harjoitukset myös tyhjälle listalle.
  + Kutsutaan loadStat-funktiota, jolle syötetään edellä muodostettu lista tilastolaskentaa varten.
* loadStat()
  + Lasketaan keskiarvot listalla olevista harjoituksista harjoitusmatkan, -ajan ja keskisykkeen osalta.
  + Syötetään nämä tiedot selaimelle, niille varatuille paikoille innerHTML-syötteenä.
  + Tehdään tarkistus mahdollisesta tyhjästä listasta ja muokataan selainnäkymää tarvittaessa, jotta vältytään null tai Nan -selainsyötteiltä.
* poistaHarjoitus()
  + Poistetaan käyttäjän valitsema harjoitus tietokannasta.
  + Kutsutaan clearAndLoad-funktiota, jotta selainnäkymän listaus ja tilastot päivittyvät.

# Yhteenveto

Pidän harjoitustyötä kokonaisuutena varsin onnistuneena. Aihe olisi tietysti voinut olla hieman kunnianhimoisempikin. Kuitenkin näkisin, että harjoitustyön aihe ja sen toteutus vastaa vähintäänkin hyvin kurssin aihealuetta ja käytössä ovat hyvällä osuudella kurssilla opittuja tekniikoita. Erityisesti pidän onnistuneena HTML-sivun skriptipuolta, joka on looginen ja hyvin selkeä. Heikkoutena ehkäpä sivuston ulkoasu, joka ei ole kovin moderni sekä tiedon esittämisen jotkin epäloogisuudet esimerkiksi listausjärjestyksen osalta. Sovelluksen selainpään kehittämiseen voisi käyttää aikaa ja mielikuvitusta vaikka kuinka paljon, mutta tässä muodossa näkisin, että ohjelman oleelliset asiat tulevat selkeästi esille.

Lähteet

Introduction to Node.js, 2023, Node.js www-sivujen esittelyosio. Viitattu 4.11.2023. https://nodejs.org/en/learn/getting-started/introduction-to-nodejs

What is MongoDB?, 03/2023, Alexander S. Gillis. Viitattu 4.11.2023. https://www.techtarget.com/searchdatamanagement/definition/MongoDB