



TAMPEREEN
AMMATTIKORKEAKOULU

OHJELMISTOTEKNIIKAN PROJEKTITYÖT, HARJOITUSTYÖ 1

Juuso Korhonen, Katri Pitkänen

Lokakuu 2016
Tietotekniikan koulutus
Ohjelmistotekniikan suuntautuminen



SISÄLLYS

1	TYÖN VAATIMUKSET	3
2	OHJELMAN ARKKITEHTUURI.....	4
3	OHJELMAN SUUNNITTELURATKAISU	5
4	OHJELMAN TOIMINTA.....	6
5	AJANKÄYTTÖ JA YHTEENVETO	7

1 TYÖN VAATIMUKSET

Harjoitustyön vaatimuksina olivat se, että sovelluksen on oltava selainpohjainen ja sen on hyödynnettävä javascriptiä ja sen päälle rakennettuja kirjastoja. Käyttäjä valitsee kaksi kartalla esitettyä suomalaista kaupunkia, ja ohjelma kertoo niiden välisen lyhyimmän reitin maanteitä pitkin. Tietokannasta löytyy tiedot kaupungeista ja niiden välisistä etäisyyksistä.

Työn tavoitteena oli tutustua uusiin web-kehitysteknologioihin ja käyttää niitä soveltavasti. Työssä painotettiin suoritusta (ongelmanratkaisua, javascriptin eri kirjastoihin tutustumista, jne.) lopputuloksen yli.

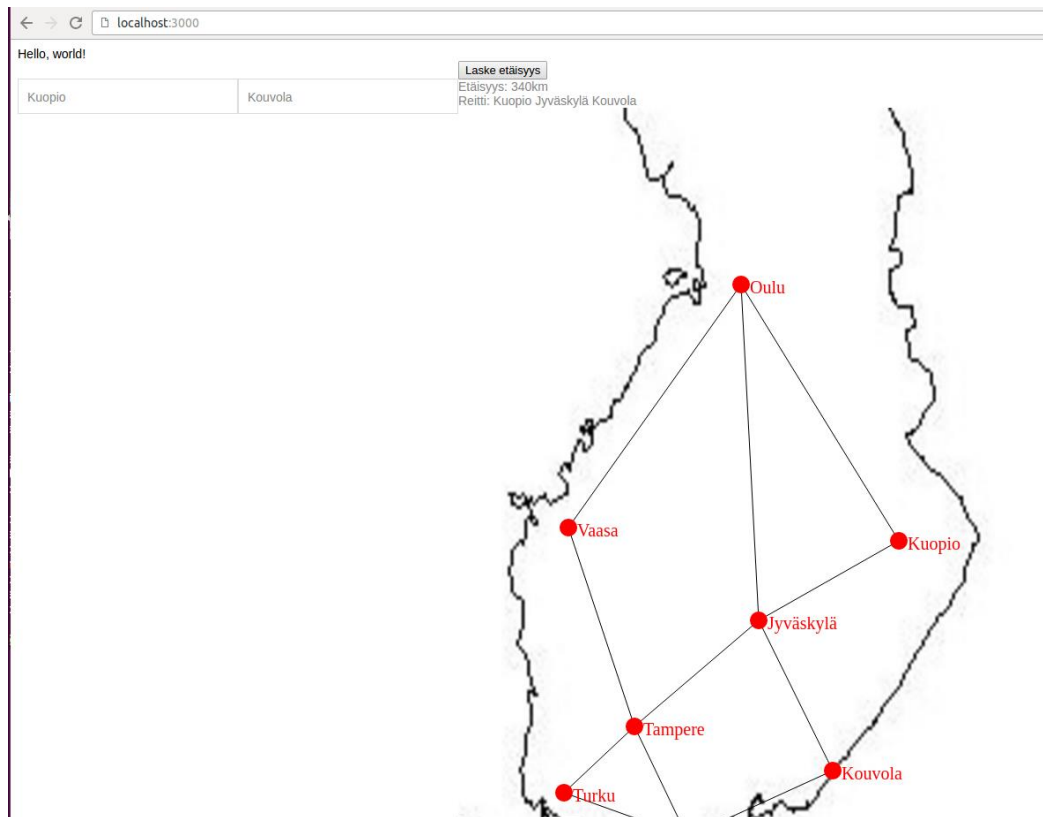
2 OHJELMAN ARKKITEHTUURI

Käytimme työssä ”MERN”-stackiä (mongodb, express.js, React.js, Node.js). MongoDB tuntui luonnollisimmalta valinnalta tietokantapalikkana enimmäkseen sen käytettävyyden takia. Työ oli pieni sovellus, joten tietokantaan ei kannattanut käyttää liikaa aikaa, koska sinne ei tarvittu juurikaan dataa. Express.js ja Node.js tulivat automaattisesti käyttöön, koska kehitimme javascriptillä, ja lopuksi, päätimme käyttää työssämme Reactia frontendinä. Reactin valitsimme sen modulaarisuuden vuoksi; sitä on helppo seurata ja laajentaa tarpeen mukaan.

3 OHJELMAN SUUNNITTELURATKAISU

Ohjelman pienestä koosta johtuen koodia ei tarvinnut juurikaan paloitella pienempiin osiin: keskeisimmät javascript-funktiot löytyvät omasta tiedostostaan, React-komponentit löytyvät omistaan, node.js:n logiikka yhdestä tiedostosta ja lopuksi kaikki renderöidään yhdelle html-sivulle. Käytimme työssä myös Githubia versionhallintaan ja npm:ää node-kirjastojen hoitoon.

4 OHJELMAN TOIMINTA



Ohjelmassa käyttäjä valitsee dropdown-listasta lähtökaupungin, toisesta listasta päätekaupungin ja painaa ”Laske etäisyys”. Tällöin javascript-funktio lähettää ajax-kutsun node.js:n REST-rajapinnalle, pyytäen etäisyyttä ja reittiä kaupunkien välillä.

Node tällöin käyttää dijkstran algoritmiä laskeakseen etäisyyden ja reitin, ja lähettää nämä tiedot responsessa takaisin javascript-funktiolle. Tämän jälkeen tiedot näytetään käyttäjälle.

5 AJANKÄYTTÖ JA YHTEENVETO

Käytimme työhön aikaa yhteensä noin 20 tuntia. 40% tästä kului ympäristön ja Reactin asentamiseen ja ymmärtämiseen + opiskeluun. Noin 20% työstä kului noden logiikan rakentamiseen, 20% näkymän piirtämiseen ja loput 20% erinäisiin irrallisiin osioihin, kuten tämän raportin tekoon, testaamiseen, versionhallintaan yms.

Työssä opimme, että MERN-stackillä jokaisella palasella on oma selkeä alueensa hoidettavana, ja ne on helppo pitää erillään oikealla ajattelutavalla. Node ja express.js hoitaa serveripään logiikan, REST-rajapinnan toteuttamisen ja keskustelee tietokannan ja käyttäjän välillä. Mongoddb hoitaa datan säilömistä ja muuttamista ja React hoitaa sisällön näyttämisen käyttäjälle.

Opimme myös, millainen on tyypillinen ohjelmointiprojekti: mitä eri vaiheita siihen kuuluu, miten sen tekemistä kannattaa jakaa. Projektin aihe oli mielenkiintoinen, ja oli hyvä, että sen toteuttamiseen sai käyttää haluamiaan (javascript) välineitä.