

Harjoitustyö

Toteutetut ominaisuudet:

- Responsiivinen ulkoasu
- Tietokannan käyttö tiedon tallennukseen
- ORM:n käyttö
- Front-controllerin käyttö
- MVC-mallin mukainen sivusto
- Front-näkymien käyttö
- Välimuistin käyttö
- FB login
- JSON:in käyttö tiedon liikuttelussa
- Grafiikka vain CSS:n avulla
- AJAX-ohjelmoinnin käyttö (axios)
- Dockerin käyttö useammalla kontilla
- Sisällöstä generoidaan pdf

Tavoiteltava pistemäärä 40-50 pistettä.

Toteutus on ajan puutteesta johtuen hieman niukka, pääasiassa viimeisenä iltana kirjoitettu. Keskityin toteutuksessa kuitenkin demonstroimaan web-kehitykseen liittyvää osaamistani mahdollisimman monipuolisesti, yksittäisten ominaisuuksien hiomisen sijaan.

Käyttöliittymä on toteutettu webpackilla, ja kehitetty pääasiassa webpack-dev-serverin avulla (hot reloading yms. näppärää) erillisessä repositoryssa, joka on liitetty harjoitustyöhön git submoduleuna. Docker-compose tekee minifoinnin myös tälle alakirjastolle, kirjastosta löytyvän `npm run build` -skriptin avulla. Kehitysympäristön saa pysytyyn tässä moduulissa `npm run serve` -komennolla.

Käyttöliittymän kehittämisessä on käytetty **bootstrap** -kirjastoa, jonka scss tiedostoista webpack kasaa Vuen single file componenteista löytyvien tyylimääritysten kanssa yhden yhteisen .css tiedoston.

Ympäristö on testattu toimivaksi Windowsilla, macOS:lla, Ubuntulla ja Archilla. Windows-ympäristössä suorittaminen vaatii dockerin ajamista linux-containereilla.

Ohjelman saa pystyyn seuraavilla komennoilla:

```
git clone git@github.com:SamiKoh/www-ht.git
cd www-ht
git submodule init && git submodule update
docker-compose up
```

Tämän jälkeen appi löytyy osoitteesta `localhost:3000`, ja mongodb:n graafinen käyttöliittymä osoitteesta `localhost:8081`.

► Kuvaus ympäristöstä

- app

- controllers
 - tietokannan käsittely
- models
 - tietokannan skeemat
- routes
 - API endpointit
- temp
 - markdown ja pdf-tiedostoja
- db
 - log
 - tietokannan lokitiedostot
 - data
 - tietokannan data