**Toteutus**

Ohjelman käyttösaatavuus, JavaScript ja halu oppia verkkotekniikoista motivoivat tekemään ohjelman verkkopohjaiseksi.

Koska ohjelmalla tuotettavien portfolioiden täytyy pystyä esittämään ääntä ja videota, päätettiin niiden muodoksi HTML-sivu, jolle tarjottaisiin kuitenkin vain lokaali tallennus mm. palvelintilan säästämiseksi.

Portfoliosivun luonti ja tallennus toteutetaan Node.js -ympäristössä.

Käyttöliittymän toteutuksen vaativampia palikoita helpotetaan jQuery-kirjastoilla.

**Kustomoidut elementit**

Portfolion ehostamisen helpottamiseksi tarjoamme käyttäjälle ns. kustomoituja elementtejä (ainakin vihjepiste (*tooltip-anchorpoint*) ja jaoteltava alue). Sisäisesti nämä ovat tavanomaisia HTML-elementtejä, joille annetaan kuitenkin ominaisuutena *data-elemType* tarkempi määritelmä, jotta editorimme mm. osaa tarjota ylimääräisiä ominaisuuksia muokattavaksi.

Vaikkakin HTML5 tarjoaa mahdollisuuden oikeasti omiin elementteihin eli omien *tagien* käyttämseen (<canvas data-elemType=’anchorpoint’> voisi olla vain <anchorpoint>), tavanomaisia elementtejä käyttämällä voidaan hyödyntää niiden omia ominaisuuksia (esim. *canvasille* piirtäminen). Omat tagit täytyy myöskin rekistöröidä JavaScriptillä, joten niiden välttäminen edesauttaa portfolio-tuloksen toimivuutta selaimissa ilman JavaScriptiä.

**Miksi tämä ei ole HTML-editori**

1. JavaScriptillä pystytään selvittämään elementin CSS-datasta vain *style*-ominaisuuden sisälle tehdyt määritteet (<div style=’margin-left:auto’>) sekä lasketut/tulostetut (*computed*) arvot.

Esim. vasen-marginaali-ominaisuuden hakeminen:

Style-määritys:

elem.style.marginLeft

Laskettu arvo:

getComputedStyle(elem).marginLeft

(jQuery: $(elem).css("margin-left"))

Mikäli margin-left -ominaisuutta ei olla elementin sisällä määritetty, ylin metodi palauttaa tyhjän merkkijonon.

Alemmat palauttavat sen sijaan aina pikseliarvon, jonka toteutus näkyy ruudulla. Tähän voi vaikuttaa vähintäänkin elementin tyyppiin, ID:hen tai johonkin luokkaan kohdistetut CSS-määritteet, joita ei voida lukea tai muuttaa ilman manuaalista CSS-tekstin käsittelyä.

Tästä syystä ohjelmalla tehdyissä/muokattavissa sivuissa **ei tule olla erillisiä CSS-tiedostoja tai yleisiä *style*-määritteitä!**

**Ohjelman kulku**

Käyttäjä valitsee haluamansa mallipohjan portfoliolle listasta, jotka ovat valmiiksi rakennettuja HTML-runkoja **tai** käyttäjä lataa ohjelmaan tallennetun HTML-sivunsa. Käyttäjän suositellaan olevan lataamatta ja muokkaamatta ohjelmalla muita kuin sillä tuotettuja sivuja, mutta tätä ei estetä.

Muokattava sivu avataan *iframe*-elementtiin. Käyttäjä voi vapaasti lisätä listalta, muokata ja poistaa sekä tavallisia että portfolion tekoa varten kustomoituja HTML-elementtejä.

Käyttäjä voi nähdä luomansa/muokkaamansa HTML-tekstin koska vain. (Mikäli HTML:n manuaalinen muokkaus sallitaan, täytyy iframessa olevan kohdesivun olla päivittymättä, kunnes muokkaus loppuu.)

Käyttäjä voi kopioida HTML:ää omaan käyttöönsä tai tallentaa sivun tiedostona omalle laitteelleen.

**Tilanne**

**KUVA TÄMÄNHETKISESTÄ OHJELMASTA**

**Mahd. ongelma**

Iframe-elementissä olevaan sivuun ei voida ainakaan Chrome-selaimessa kohdistaa muokkauksia, elleivät sivu ja editori sijaitse samassa kohteessa, *domainissa.* Tällöin muokattava sivu saatetaan joutua tuomaan aina ensin palvelimelle, mikä saattaa olla tietoturvariski.