



LAHDEN AMMATTIKORKEAKOULU
Lahti University of Applied Sciences

WWW-SIVUSTON JATKOKEHITYS

Case: DynaStart

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikka
Ohjelmistotekniikka
Opinnäytetyö
Syksy 2011
Teemu Jukko

Lahden ammattikorkeakoulu
Tietotekniikka

JUKKO, TEEMU:

WWW-sivuston jatkokehitys
Case: DynaStart

Ohjelmistotekniikan opinnäytetyö, 44 sivua

Syksy 2011

TIIVISTELMÄ

Opinnäytetyön taustalla on tarve saada nykyinen sivustototeutus uudelleen määriteltyjen asiakasvaatimusten mukaiseksi. Kyseessä on ensimmäisen toteutuksen jatkokehitys, ja tavoitteena on saada sivusto julkaisukelpoiseksi syksyn 2011 aikana.

Edellinen toteutus kulki nimellä Dyna Startup, ja se tarjosi mahdollisuuden yrittäjähenkiseen ideoiden esilletuontiin ja niiden tukemiseen. Jatkokehityksen ideana oli uudistaa sivuston ulkoasu sekä muokata ja luoda uusia sivuston toimintoja. Sivuston käyttötarkoitus pysyy samana, ja oleellisimman osan muodostavat käyttäjien luomat projektit.

Teoriaosuus keskittyy sivuston asiakkaan ja palvelimen väliseen kommunikaatioon ja sen tekniikoihin. Esille nostetaan ennen kaikkea vapaassa jakelussa oleva JavaScript-kirjasto jQuery ja tuodaan esille sen käyttötarkoitukset ja -mahdollisuudet. JQueryn keskeisimpiä käyttötarkoituksia ovat työn kannalta asynkroniset kutsut, jotka mahdollistavat työpöytämaisemmän kokemuksen palvelua selattaessa ja käytettäessä.

Toteutus eteni haluttujen vaatimusten mukaisesti, ja keskeisimpiä toteutettavia kokonaisuuksia priorisoitiin aikataulun niin vaatiessa. Edellisen toteutuksen tavoin, sivusto pyrkii tarjoamaan käyttäjälleen mahdollisimman käyttäjäystävällisen lopputuloksen Ajaxia ja dynaamisuutta hyödyntäen.

Lopputuloksena sivusto saatiin enemmän siihen muotoon, kun sen asiakas oli alunperinkin halunnut ja toivonut. Ulkoasu uudistui, sekä käyttäjälle toteutettiin uusia toimintoja ja vaikutusmahdollisuuksia. Sivuston uudeksi nimeksi tuli DynaStart. Jatkon osalta sivustoa tullaan edelleen kehittämään uusien ominaisuuksien ja toiveiden muodossa.

Avainsanat: Ajax, dynaaminen, jatkokehitys, jQuery

Lahti University of Applied Sciences
Degree Programme in Information Technology

JUKKO, TEEMU:

Website development
Case: DynaStart

Bachelor's Thesis in Software Engineering, 44 pages

Autumn 2011

ABSTRACT

The purpose of this thesis was to modify the current implementation of the website of DynaStart according to the new customer requirements. This was the first development from the initial implementation and the goal was to get the site published in autumn 2011.

The previous implementation was called Dyna Startup. It was a site where the users could create and support entrepreneurial ideas. The goal with the new site was to rebuild the appearance of the site and also modify and create new functions to the site. The purpose of the site will remain the same and the essential parts are the projects created by the users of the site.

The theory part focuses on communication between the client and the server and its techniques. The main focus is on the lightweight JavaScript library called jQuery along with its purposes and possibilities. As for the site development, one of the key parts of using the jQuery are asynchronous calls, which provide the user with more desktop oriented experience when browsing and using the site.

The site development proceeded according to the customer requirements and the essential functions were prioritized when the schedule required. Like the previous implementation, the site aims to provide a user friendly experience to its user through Ajax and dynamic content.

As a result, the site was implemented to the form that the customer had planned and wished in the first place. The appearance of the site changed and current functions and new functions were implemented. The name of the site changed to DynaStart. As for the future, the site development will proceed with new features and functions.

Key words: Ajax, dynamic, development, jQuery

SISÄLLYS

1	JOHDANTO	1
2	ENSIMMÄINEN TOTEUTUS JA ASIAKASVAATIMUKSET	2
3	KÄYTETYT TEKNIIKAT	4
3.1	HTTP-protokollan toiminta	4
3.2	HTML DOM	7
3.3	Jquery	10
3.3.1	Yleistä	10
3.3.2	Valitsijat	11
3.3.3	Elementtien käsittely	12
3.3.4	Lisäosat	12
3.3.5	Jquery UI	14
3.4	AJAX	16
3.5	MySQL Event Scheduler	18
3.5.1	Yleistä	18
3.5.2	Toiminta	19
4	TOTEUTUS	21
4.1	Tietokantarakenne	21
4.1.1	Muutokset nykyisiin tauluihin	22
4.1.2	Uudet taulut	23
4.2	Sivutus	25
4.3	Ennakoiva haku	28
4.4	Käyttäjätili	30
4.4.1	Krediittijärjestelmä	32
4.4.2	Muokkaus	34
4.4.3	Salasana unohtunut	35
4.4.4	Poisto	36
4.5	Projekti	37
4.5.1	Luonti	37
4.5.2	Muokkaus	39
4.5.3	Sosiaalinen media	41
4.5.4	Poisto	42
4.6	Kielivalinta	42

5 YHTEENVETO

44

LÄHTEET

45

1 JOHDANTO

Työn taustalla on asiakkaan tarve saada nykyinen sivustototeutus haluamaansa ulkonäköön ja toiminnallisuuteen, eli toisin sanoen asiakasvaatimusten mukaiselle tasolle. Asiakkaana on Dyna, joka on Päijät-Hämeen koulutus konsernin alaisuudessa toimiva monialainen yrittäjyysfoorumi, joka tarjoaa yrittäjyyteen liittyviä palveluita ja ohjaa henkilön tarvittaessa eteenpäin.

Sivusto itsessään liittyy niin ikään yrittäjyyteen ja kulkee tätä nykyä nimellä DynaStart, edellisen toteutuksen ollessa puolestaan Dyna Startup. Lähtökohta on kuitenkin pysynyt samana. Käyttäjä rekisteröityy sivulle, minkä jälkeen on mahdollista luoda omia projekteja sekä tukea ja kommentoida muiden käyttäjien projekteja.

Muutosten myötä käyttäjälle haluttiin ensisijaisesti antaa enemmän toiminnallisuutta käyttökohteiden osalta ja saada tietty logiikka projektien tukemiseen. Pohjimmainen kysymys oli, voiko nämä hahmotellut muutokset toteuttaa onnistuneesti jatkaen nykyistä toteutusta. Teoriaosuudessa tutustutaan web-maailman yleisiin käytäntöihin sekä otetaan syvempään tarkasteluun tekniikat ja teknologiat, joihin rakennusvaiheessa on päädytty ja ne myös toimivaksi todettu.

2 ENSIMMÄINEN TOTEUTUS JA ASIAKASVAATIMUKSET

Sivustosta tehty ensimmäinen toteutus tehtiin niin ikään opinnäytetyönä, ja se kulki nimellä Dyna Startup. Sivuston ideana oli yrittäjähenkisen ideoiden esiintuonti ja niiden tukeminen. Perusominaisuudet muodostuivat palveluun rekisteröitymisestä, projektien tukemisesta ja selailusta sekä erillisestä ylläpitäjälle suunnatusta hallintapaneelistä. Vastaavia palveluja on esimerkiksi Kickstarter, jota ideointivaiheessa hyödynnettiin. Useimmista muista poiketen, Suomen lainsäädännöllisten syiden takia Dyna Startup ei käytä tukemiseen oikeaa rahaa, vaan käytössä on virtuaalisia krediittejä, joiden yksikkö on D.

Uusi toteutus on nimeltään DynaStart, joka hahmottui nimen osalta projektin edetessä. Toteutuksesta vastasivat yksi visualistiopiskelija sekä ohjelmistotekniikan opiskelija. Sivustoa rakennettiin asiakkaan tiloissa. Käytössä oli GoogleDocs-työkalu, joka mahdollisti reaaliaikaisen yhteydenpidon ja ajatusten vaihdon dokumenttimuodossa.

Asiakasvaatimukset selkeyttivät sivuston toimintaa. Ensimmäisestä toteutuksesta poiketen, käyttäjällä on rajallinen määrä krediittejä käytettävään kuukautta kohden. Päivän ensimmäisestä sivustolle kirjautumisesta palkitaan lisäkrediiteillä. Nämä toistuvat toiminnallisuudet toivat tarpeen ajastetuista toiminnoista, jotka suoritetaan tietyin väliajoin. Käyttäjällä on myös oltava enemmän toimintoja käytettävään sivustolla. Omia projektejaan ja käyttäjätietojaan tulee voida muokata ja poistaa, sekä on myös mahdollista vaihtaa unohtunut salasana sähköpostiin tulevan linkin kautta. Omille projekteille ja tukemishistorialle haluttiin myös oma näkymä. Kommentoinnin osalta on mahdollista vastata johonkin jo aloitettuun kommenttiin tai kommenttiketjuun.

Projektien ja käyttäjien määrän lisääntyessä sivustolle haluttiin myös sivustusosioita eri tilanteita varten. Projektien eri listauksille tehtiin numerosivutukset, joista käyttäjä voi sivunumeroa painamalla saada lisää projekteja näkyviin. Sivutus otettiin käyttöön myös käyttäjien antamissa kommentteissa ja tukijoita ja blogeja selatessa.

Kansainvälisyyttä silmällä pitäen tarvetta oli sivuston käyttökielen valinnalle. Tarvetta oli sekä suomenkieliselle että englanninkieliselle toteutukselle. Pääkäyttäjryhmä muodostuu suomenkielisistä käyttäjistä, joten suomen kieli valittiin sivuston ensisijaiseksi kieleksi.

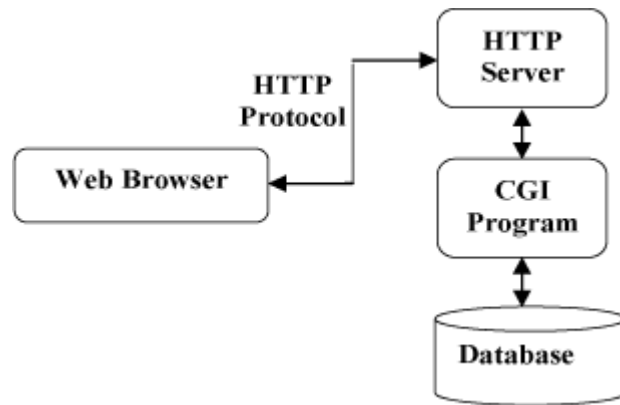
3 KÄYTETYT TEKNIIKAT

3.1 HTTP-protokollan toiminta

HTTP, joka on lyhenne sanoista Hypertext Transfer Protocol, on TCP/IP:hen perustuva kommunikointiprotokolla, jota käytetään tietojen eli resurssien välittämiseen www-ympäristössä. Nämä kollektiiviset resurssit voivat olla esimerkiksi kuvatiedostoja, kyselytuloksia tai HTML-tiedostoja. (Tutorials Point 2011.)

Käyttäjän selain (esimerkiksi Mozilla Firefox tai Internet Explorer) toimii HTTP-asiakkaana, koska se lähettää pyyntöjä HTTP-palvelimelle (web-palvelin). Tämä web-palvelin lähettää tämän jälkeen vastauksia takaisin asiakkaalle. HTTP-palvelimien kuunteluportti on oletuksena 80, mutta se voidaan muuttaa myös muuksi arvoksi. (Tutorials Point 2011.)

Kuviossa 1 on havainnollistettu HTTP-protokollan toimintaa ja sijoittumista. Käytettävänä ratkaisuna voivat olla esimerkiksi selain Mozilla Firefox (Web Browser), web-palvelin Apache (HTTP Server), komentosarjakieli PHP (CGI Program) ja relaatiotietokantaohjelmisto MySQL (Database) tiedon varastointiin. Asiakkaan selain lähettää pyynnön web-palvelimelle, joka palauttaa vastauksen asiakkaalle. CGI, joka muodostuu sanoista Common Gateway Interface, on kieliriippumaton Web-ympäristön tekniikka, jonka avulla selain voi välittää tietoa palvelimella ajettavalle ohjelmalle. CGI voi myös itsessään tarkoittaa ajettavaa ohjelmaa (cgi-päätteinen tiedosto). (Wikipedia 2011.)



KUVIO 1. HTTP-protokollan toiminta (Tutorials Point 2011)

HTTP-protokollan ominaisuuksiin kuuluu seuraavaa:

- Yhteydettömyys, eli kun pyyntö on tehty, niin asiakas katkaisee yhteyden palvelimelta ja odottaa vastausta. Palvelimen pitää muodostaa yhteys uudelleen käsiteltyään pyynnön.
- Tiedon itsenäisyys, eli data voi olla minkä muotoista tahansa, kunhan sekä asiakas- että palvelinpuoli osaa käsitellä tätä sisältöä.
- Tilattomuus, joka on seuraus yhteydettömyydestä. Palvelin ja asiakas ovat tietoisia toisistaan ainoastaan pyyntöä tehdessä, ja ne unohtavat toisensa tämän jälkeen.

(Tutorials Point 2011.)

Kuten useimmat verkkoprotokollat, HTTP käyttää asiakas-palvelin -mallia, jossa HTTP-asiakas avaa yhteyden ja lähettää pyynnön HTTP-palvelimelle. Palvelin palauttaa tämän jälkeen vastauksen, joka yleensä sisältää pyydetyn resurssin. Vastauksen myötä palvelin sulkee yhteyden. (Tutorials Point 2011.)

Kuviossa 2 on esimerkki HTTP-pyyntöstä. Ensimmäinen rivi sisältää käytettävän metodin (GET), pyydetävän resurssin (/path/file.html) ja HTTP-protokollan version (1.0). Toinen ja kolmas rivi ovat pyynnön header-rivejä, jotka tarjoavat tietoa pyynnöstä. Ensimmäinen header-kenttä on "From", jonka tulisi olla sähköpostiosoite käyttäjälle, joka hallinnoi pyyntöä tekevää asiakasohjelmaa. Toinen header-kenttä "User-Agent" sisältää tietoa pyyntöä tekevästä asiakasohjelmasta. Tämän jälkeen tulee tyhjä rivi. (Tutorials Point 2011.)

```

1 GET /path/file.html HTTP/1.0
2 From: someuser@tutorialspoint.com
3 User-Agent: HTTPTool/1.0
4 [blank line here]

```

KUVIO 2. HTTP-pyyntö GET-metodilla (Tutorials Point 2011)

Kuviossa 3 on esimerkki HTTP-vastauksesta. Ensimmäinen rivi (status line) sisältää kolme tietoa: käytetyn HTTP-protokollan version (1.0), tilakoodin (200) joka antaa pyynnön vastauksen ja kolmantena on kyseistä tilakoodia englanniksi kuvaava lauseke (OK). 1xx-alkuiset tilakoodit kuvaavat tietoa, 2xx-alkuiset onnistumista, 3xx-alkuiset uudelleenohjaamista, 4xx-alkuiset asiakaspuolen virhettä ja 5xx-alkuiset palvelinpuolen virhettä. Seuraavat kolme riviä sisältävät vastauksen header-osion. ”Date” kertoo päiväyksen ja ajan, jolloin viesti sai alkunsa (Fri, 31 Dec 1999 23:59:59 GMT), ”Content-Type” kuvaa lähetetyn sisällön mediatyyppiä (text/html) ja ”Content-Length” kertoo lähetetyn sisällön koon tavuina (1354). Loppuosa on viestin sisältö (body), joka palautetaan vastauksena asiakkaalle. (Tutorials Point 2011.)

```

1 HTTP/1.0 200 OK
2 Date: Fri, 31 Dec 1999 23:59:59 GMT
3 Content-Type: text/html
4 Content-Length: 1354
5
6 <html>
7 <body>
8 <h1>Happy New Millenium!</h1>
9
10 (more file contents)
11 .
12 .
13 .
14 </body>
15 </html>

```

KUVIO 3. HTTP-pyyntön vastaus (Tutorials Point 2011)

GET-metodissa siis noudetaan mikä tahansa entiteettimuotoinen tieto, joka tunnistetaan resurssin perusteella. Toinen oleellinen metodi on nimeltään POST, jota käytetään tiedon lähettämiseen palvelimelle käsiteltäväksi. POST-pyyntö eroaa GET-pyyntöstä seuraavin tavoin:

- Pyyntön kanssa lähetetään lohko, joka sisältää tietoa viestin sisällössä.

- Pyynnön kohde on lähes poikkeuksetta ohjelma, joka käsittelee lähetetyn tiedon.
- Pyynnön vastaus on yleensä ohjelman lopputulos (output) eikä staattinen tiedosto.

Yleisin POST-metodin käyttökohde on lähettää HTML-lomakkeen tieto sitä prosessoivalle ohjelmalle. Kuviossa 4 on esimerkki lomakkeen lähetyksestä POST-metodia käyttäen, jossa tietoina kulkevat home (arvo ”Mosby”) ja favorite flavor (arvo ”flies”). (Tutorials Point 2011.)

```

1 POST /path/script.cgi HTTP/1.0
2 From: frog@jmarshall.com
3 User-Agent: HTTPTool/1.0
4 Content-Type: application/x-www-form-urlencoded
5 Content-Length: 32
6
7 home=Mosby&favorite+flavor=flies

```

KUVIO 4. Lomakkeen lähetyks POST-metodilla (Tutorials Point 2011)

3.2 HTML DOM

Yleisellä tasolla puhuttaessa DOM (Document Object Model) on W3C-säätiön (World Wide Web Consortium) alusta- ja kieliriippumaton rajapinta. Se mahdollistaa ohjelmien ja skriptien päästä dynaamisesti käsiksi ja muokkaamaan jotakin dokumentin sisältöä, rakennetta ja tyyliä. DOM sisältää kolme eri osaa tai tasoa: ydin-DOM, XML DOM ja HTML DOM. Näistä HTML DOM määrittelee HTML-elementtien oliot ja ominaisuudet sekä metodit niiden käsiksi pääsyyn. Toisin sanoen, HTML DOM on standardi jolla HTML-elementtejä otetaan käsittelyyn, muokataan, lisätään ja poistetaan. (W3Schools 2011b.)

Puhuttaessa HTML DOM:sta kaikki HTML-dokumentissa ovat solmuja (node). Koko dokumentti on dokumenttisolmu, HTML-elementti on elementtisolmu, HTML-elementtien sisältämä teksti on tekstisolmu, HTML-attribuutti on attribuuttisolmu ja kommentit ovat kommenttisolmuja. Kuviossa 5 on esimerkki HTML-dokumentista. Siinä juurisolmuna (root node) toimii html. Tällä juurisolmulla on kaksi lapsisolmua: head ja body. Head-solmulla on puolestaan title-solmu, ja niin edelleen. (W3Schools 2011b.)

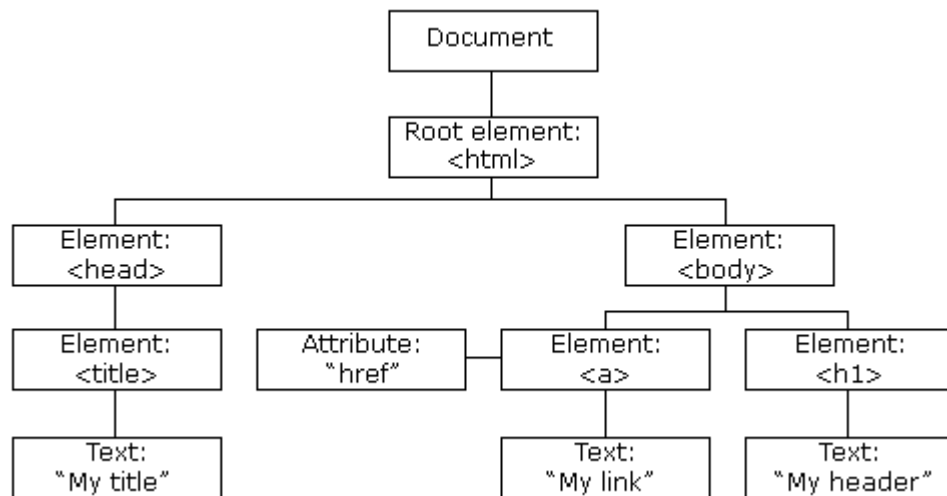
```

1  <html>
2  <head>
3      <title>My title</title>
4  </head>
5  <body>
6      <h1>My header</h1>
7      <a href="http://www.w3schools.com">My link</a>
8  </body>
9  </html>

```

KUVIO 5. HTML-dokumentti (W3Schools 2011)

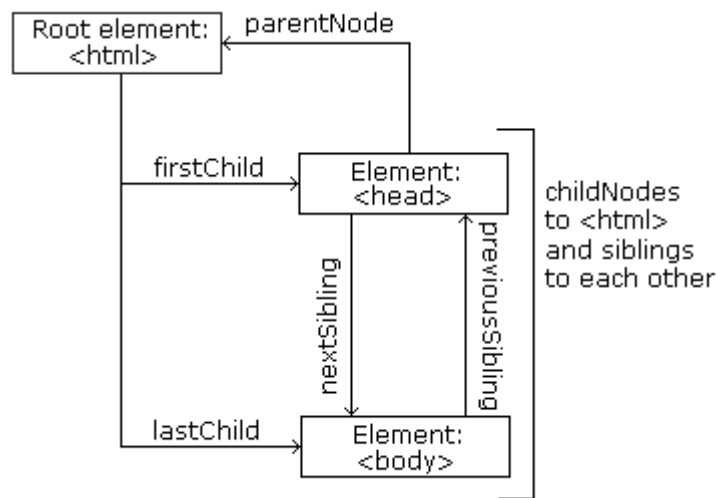
HTML DOM tarkastelee HTML-dokumenttia puurakenteena, joka on nimeltään solmupuu (node-tree). Jokaiseen solmuun päästään käsiksi solmupuun kautta, ja niiden sisältöä voidaan muokata ja poistaa, sekä lisäksi luoda uusia elementtejä. Kuviossa 6 on kuvattu kuvion 5 HTML-dokumentin solmupuu, jossa on solmut ja niiden yhteydet. (W3Schools 2011b.)



KUVIO 6. Esimerkki solmupuusta (W3Schools 2011)

Solmupuun solmuilla on hierarkinen yhteys toisiinsa. Näitä yhteyksiä kuvataan termeillä ”parent” (emo), ”child” (lapsi) ja ”sibling” (sisar). Tämä tarkoittaa sitä, että emოსolmuilla on lapsisolmuja, ja saman tason lapsisolmut ovat sisaruksia. Solmupuun ensimmäinen solmu on nimeltään juurisolmu (root node). Jokaisella solmulla (poislukien juurisolmu) on täsmälleen yksi emოსolmu. Lapsisolmujen määrää ei ole rajattu. Jos solmulla ei ole lapsia, käytetään termiä ”leaf” (lehti).

Kuviossa 7 on kuvattu solmupuun osaa ja siinä esiintyvien solmujen yhteyksiä. (W3Schools 2011b.)



KUVIO 7. Solmupuun solmujen yhteyksiä (W3Schools 2011)

DOM:sta puhuttaessa HTML-dokumentit muodostuvat solmuolioista (node objects). Näihin solmuihin päästään käsiksi esimerkiksi JavaScript-komentosarjakielen avulla. DOM-ohjelmointirajapinta määrittellään ominaisuuksien ja metodien mukaan. Ominaisuuksilla viitataan useasti johonkin, mikä joku on, kuten esimerkiksi solmun nimi. Näitä HTML DOM -ominaisuuksia ovat esimerkiksi (x on tarkasteltava HTML-elementti, solmuolio):

- x.innerHTML - x:n tekstiarvo
- x.nodeName - x:n nimi
- x.nodeValue - x:n arvo
- x.parentNode - x:n emოსolmu
- x.childNodes - x:n lapsisolmu(t)
- x.attributes - x:n attribuutit.

Metodeilla puolestaan viitataan useasti johonkin mitä tehdään, kuten esimerkiksi poistetaan jokin solmu. HTML DOM -metodeja ovat esimerkiksi (x on tarkasteltava HTML-elementti, solmuolio):

- x.getElementById(id) - antaa määritellyn id:n elementin
- x.getElementsByTagName(name) - antaa kaikki elementit määritellyn nimen (tag name) perusteella

- `x.appendChild(node)` - lisää määritellyn lapsisolmun `x`:ään
- `x.removeChild(node)` - poistaa määritellyn lapsisolmun `x`:stä.

(W3Schools 2011b.)

Jokaisella HTML-sivun elementillä on lisäksi tapahtumia (events), joita voidaan käyttää JavaScriptin avulla. Esimerkkejä näistä tapahtumista ovat muun muassa:

- hiiren painaminen ja siirtäminen
- kuvan lataus
- lomakkeen lähetys
- näppäinpainallus.

Tapahtumat määritellään HTML-elementeissä ja niitä käytetään normaalisti yhdessä jonkin itsemääritellyn funktion kanssa, joka suoritetaan tapahtuman toteuduttua. Kuviossa 8 on esimerkki `onchange`-tapahtumasta `input`-elementissä, joka toteutuessaan laukaisee `checkEmail`-funktion suorituksen. Tässä tapauksessa tapahtuma toteutuu, kun käyttäjä vaihtaa tämän `input`-kentän sisältöä. (W3Schools 2011b.)

```

1  <html>
2  <head>
3  .
4  .
5  </head>
6  <body>
7  .
8  .
9  E-mail: <input type="text" id="email" onchange=checkEmail () " />
10 .
11 .
12 </body>
13 </html>

```

KUVIO 8. Onchange-tapahtuma `input`-kentässä (W3Schools 2011)

3.3 JQuery

3.3.1 Yleistä

Jquery on kooltaan kevyt kirjasto JavaScript-komentosarjakiellelle. Sen ominaisuuksiin kuuluu muun muassa seuraavaa:

- HTML-elementtien valitsijat

- HTML-elementtien muokkaus
- CSS-tyylimäärittelyjen muokkaus
- HTML-tapahtumatoiminnot (event functions)
- JavaScript-efektit ja animaatiot
- HTML:n DOM-puurakenteessa liikkuminen ja muokkaus
- AJAX
- Apuohjelmat (lisäosat).

(W3Schools 2011c.)

Jquery tarjoaa ratkaisun pitää kirjoitettava koodi yksinkertaisena ja suppeana. Tämä kirjasto on myös suunniteltu siten, että asioita voidaan helposti käyttää uudelleen. Jquery yksinkertaistaa HTML DOM -puurakenteessa liikkumisen ja se tarjoaa lisäksi useita helppokäyttöisiä toimintoja (funktioita) rikkaiden internetsovellusten tekemiseen. Käyttö on mahdollista lähes kaikkien muiden web-ohjelmointikielien kanssa, kuten JSP:n, ASP:n ja PHP:n. (Rose India 2008.)

3.3.2 Valitsijat

Jquery mahdollistaa elementtien tai elementtiryhmiä hallinnoinnin HTML DOM -puurakenteessa CSS- ja XPath-valitsijoiden avulla. Valitsijoilla saatu tulos on jQuery-oliot, jotka ovat erilaisia DOM-elementteihin verrattuna. Näihin jQuery-olioihin voidaan sijoittaa tapahtumia ja lisätä efektejä sekä ketjuttaa useita muutoksia tai efektejä toisiinsa. Valitsijoita käytettäessä syntaksi aloitetaan useasti dollarimerkillä (\$) ja tämän jälkeen haluttu valinta tulee sulkumerkkeihin. Tämä toiminnallisuus poistaa tarpeen käydä jokainen ryhmäelementti läpi, sillä kaikki, mitä sulkujen sisään laitetaan, käydään läpi automaattisesti ja varastoidaan jQuery-olioksi. (Chaffer & Swedberg 2007, 17 - 18.)

Taulukossa 1 on esimerkkejä HTML-elementtien valitsemisista ja kuvaus, mihin ne pätevät. HTML-elementtien attribuutteihin viitattaessa käytetään XPath-valitsijaa, ja HTML-elementteihin viitattaessa käytetään CSS-valitsijaa. jQueryllä on myös CSS:n tavoin olemassa omia custom-valitsijoita, joita ovat esimerkiksi taulukossa 1 esiintyvä first. (Chaffer & Swedberg 2007, 24.)

TAULUKKO 1. Esimerkkejä valitsijoista jQuerylla

Syntaksi	Kuvaus
<code>\$(this)</code>	Nykyinen HTML-elementti
<code>\$("p")</code>	Kaikki <code><p></code> -elementit
<code>\$("#p.intro")</code>	Kaikki <code><p></code> -elementit joilla luokka "intro"
<code>\$("#p#intro")</code>	Kaikki <code><p></code> -elementit joilla id "intro"
<code>\$("#p.intro:first")</code>	Ensimmäinen <code><p></code> -elementti jolla id "intro"
<code>\$(".intro")</code>	Kaikki elementit joilla luokka "intro"
<code>\$("#intro")</code>	Ensimmäinen elementti jolla id "intro"
<code>\$("#ul li:first")</code>	Jokaisen <code></code> -elementin ensimmäinen <code></code> -elementti
<code>\$("#[href\$='.jpg']")</code>	Kaikki elementit joilla href-attribuutti, joka loppuu ".jpg"
<code>\$("#div#intro.head")</code>	Kaikki elementit id:n "intro" omaavan <code><div></code> -elementin sisällä luokalla "head"

3.3.3 Elementtien käsittely

jQuery mahdollistaa HTML-elementtien ja -attribuuttien vaihtamisen ja manipuloinnin metodien (funktioiden) avulla. Elementtejä voidaan kopioida, siirtää tai luoda tyhjästä. Jokaiseen elementtiin tai elementtiryhmään on mahdollista liittää ja kietoa (wrap) muita elementtejä. (W3Schools 2011c; Bibeault & Katz 2008, 81.)

Yksi tapa elementtiryhmiä muokkaamiseen on `each`-metodi. Kuviossa 9 on esimerkki `img`-elementtien `alt`-attribuutin muokkaamisesta `each`-metodia käyttäen. Jokainen sivulta löytyvä `img`-elementti saa `alt`-arvokseen määritellyn tekstin, jossa on muuttuvina osina kuvan numero (alkaa nollostasta) ja kuvan `id`-attribuutti. `Ready`-metodilla varmistetaan, ettei dokumentin elementtejä muokata ennen kuin ne ovat latautuneet. (Bibeault & Katz 2008, 51.)

```

1  $(document).ready(function() {
2      $('img').each(function(n) {
3          this.alt='This is image['+n+'] with an id of '+this.id;
4      });
5  });

```

KUVIO 9. `Each`-metodin käyttö elementtiryhmään (Bibeault & Katz 2008)

3.3.4 Lisäosat

jQuery tarjoaa työkalun, jossa on käytettävänä komentoja ja toimintoja, joita voidaan yhdistää toisiinsa ja antaa sivuille haluttu toiminnallisuus. Kun koodi

käyttää yleisiä malleja näihin toimintoihin, niitä käytetään uudelleen ja uudelleen. Näiden mallien myötä on aihetta tehdä nämä toiminnot erilliseksi, uudelleenkäytettäväksi työkaluksi jQueryn rinnalle. JQueryn lisäosat (plugins) mahdollistavat käyttäjän hyödyntämään jQuery-kirjastoa tehokkaammin ja vetämään yhteen hyödylliset funktiot uudelleenkäytettäväksi koodiksi, jonka tavoitteena on säästää aikaa ja mahdollistaa kehityksen sujuvuus. (Bibeault & Katz 2008, 186; The jQuery Project 2010.)

Yksi näistä lisäosista on nimeltään Validation, joka tekee asiakaspuolen lomakkeen tarkastuksesta joustavaa, tarjoten samalla paljon räätälöintimahdollisuuksia. Validation-lisäosan käyttö on mahdollista sekä uuden että nykyisen web-sivuston osalta. Lisäosa tukee useita tarkastusmetodeja, kuten URL- ja email-osoitteen validoinnin, sekä myös tarjoaa rajapinnan omien metodien kirjoittamiseen. Lisäosan virheilmoitusviestit on mahdollista saada englanniksi sekä 36 muulle kielelle. (Zaefferer 2009.)

Kuviossa 10 on esimerkki Validation-lisäosan käytöstä. Tämän lomakkeen id on commentForm, johon kohdistetaan JavaScript-koodissa Validate-lisäosan validate-funktio. Virhetilanne on syntynyt lomakkeen lähetyksnappia (Submit) painettaessa. Tähdellä merkittyihin kenttiin on HTML-koodissa annettu luokka "required", ja virhetilanteen tapahtuessa niille lisätään lisäksi luokka "error". Lisäksi muodostaan label-elementti virheellisen kentän perään, jossa kerrotaan käyttäjälle virheestä. Kuviossa 11 on havainnollistettu dokumentin HTML-rakennetta virheellisen nimikentän osalta. Toimintaan kuuluu myös se, että lomaketta ei virhetilanteessa lähetetä palvelimelle prosessoitavaksi, vaan estetään sen normaali toiminta. (The jQuery Project 2010.)

```
$("#commentForm").validate();
```

A simple comment form with submit validation and default messages

Name * This field is required.

E-Mail * This field is required.

URL

Your comment * This field is required.

KUVIO 10. Validate-lisäosan käyttö HTML-lomakkeessa (The jQuery Project 2010)

```

1 <html>
2 <head>
3 .
4 .
5 </head>
6 <body>
7 .
8 .
9 <form class="cmxform" id="commentForm" method="get" action="">
10 .
11 <input id="cname" name="name" size="25"
12 class="required error" minlength="2" />
13 <label for="cname" generated="true"
14 class="error">This field is required.</label>
15 .
16 </form>
17 .
18 .
19 </body>
20 </html>

```

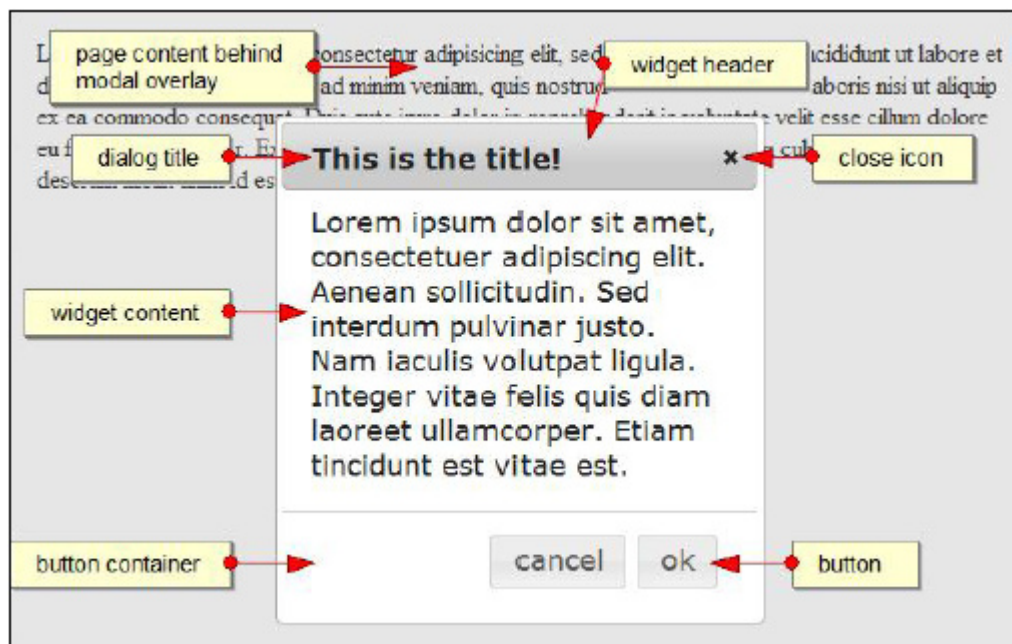
KUVIO 11. HTML DOM - virheellinen nimikenttä lomakkeessa (The jQuery Project 2010)

3.3.5 Jquery UI

Jquery UI on virallinen käyttöliittymäkirjasto jQuerylle. Jquery UI laajentaa jQuery-kirjastoa tarjoamalla sarjan interaktiivisia käyttöliittymätyökaluja (widgets), joilla parannetaan web-sivujen ja -sovellusten käyttöliittymiä. (Wellman 2011, 7.)

Jquery UI:n käyttö perustuu jQueryyn, ja syntaksi komponenttien alustamiseen, konfigurointiin ja manipulointiin tapahtuu kuten jQueryssa. Tämän kirjaston tarkoituksena on helpottaa käyttöliittymien rakentamista. Jokainen komponentti on suunniteltu olemaan tehokas, pienikokoinen ja semanttisesti oikea, sekä tarjoten JavaScriptin uusimmat olio-ominaisuudet käyttöön. (Wellman 2011, 8, 23.)

Yksi jQuery UI:n widgeteistä on ”dialog”, joka mahdollistaa tiedon (esimerkiksi viestin tai kuvan) näyttämisen. Siihen on myös helppo lisätä nappeja, kuten esimerkiksi ok- ja peruuta-napit. Näihin nappeihin voi määrittellä niin kutsuttuja callback-funktioita, joilla ne saadaan reagoimaan, kun käyttäjä painaa niitä. Dialog-widgetti voi olla modaalinen tai ei-modaalinen, ja sillä on lisäksi erilaisia säätömahdollisuuksia. Yksi näistä valinnoista on ”autoOpen”, jonka oletusarvo on ”true”. Arvolla ”true” dialogi näytetään heti kun dialog-metodia kutsutaan. Jos haluttaisiin määrittellä dialogin avautuminen esimerkiksi hiirenpainalluksen yhteydessä, tulisi arvoksi laittaa ”false” ja tehdä erillinen click-metodi jQuerylla. Kuviossa 12 on esimerkki dialog-widgetistä. (Wellman 2011, 113, 117.)



KUVIO 12. Dialog-widget -esimerkki (Wellman 2011)

3.4 AJAX

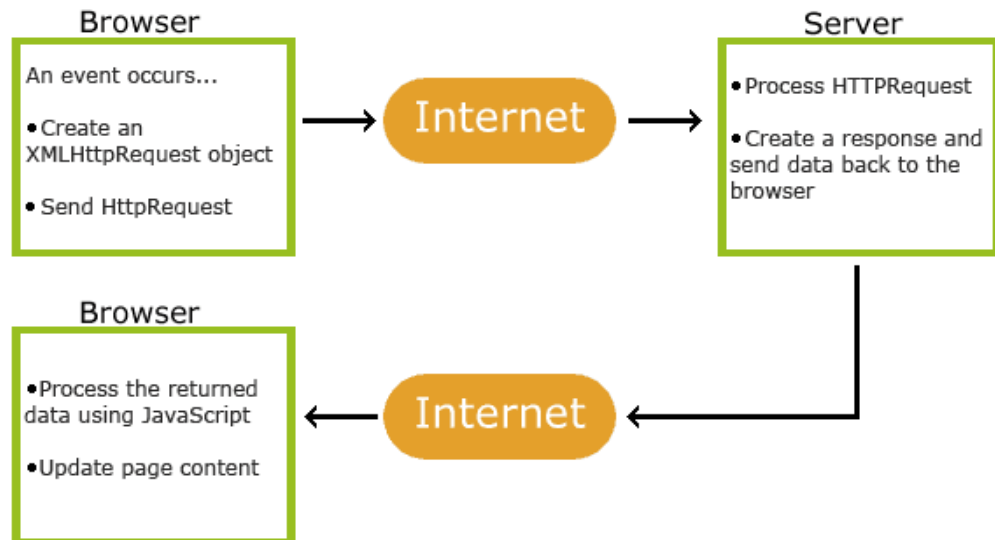
AJAX (Asynchronous JavaScript and XML) on selain- ja alustariippumaton teknologia nopeiden ja dynaamisten web-sivujen luontiin. Se ei itsessään ole ohjelmointikieli, vaan tapa käyttää käytössä olevia standardeja. Se muodostuu seuraavista standardeista:

- XMLHttpRequest-olio, joka vaihtaa tietoa asynkroonisesti palvelimen kanssa
- JavaScript/DOM, joka mahdollistaa tiedon näyttämisen ja käytön
- CSS, jolla tyylitetään tieto
- XML, jota usein käytetään liikutettavan tiedon formaattina.

(W3Schools 2011a.)

AJAX mahdollistaa web-sivujen asynkroonisen päivityksen vaihtamalla pieniä määriä tietoa palvelimen kesken. Kyseinen vaihto tapahtuu käyttäjältä piilossa. Tämä tarkoittaa sitä, että on mahdollista päivittää sivun osia ilman sen uudelleenlataamista. Perinteisessä web-sivussa sisällön vaihtuessa tapahtuu sivulataus. Esimerkkisovelluksia, jotka hyödyntävät AJAX-teknologiaa, ovat esimerkiksi Google Maps, Gmail, Youtube ja Facebook. (W3Schools 2011a.)

Kuviossa 13 on havainnollistettu AJAX:n toimintaa. Tapahtuman (esimerkiksi hiiren painallus) tapahduttua muodostetaan XMLHttpRequest-olio, joka lähetetään palvelimelle käsiteltäväksi esimerkiksi GET-metodia käyttäen. Palvelin muodostaa vastauksen ja lähettää tiedon takaisin asiakkaan selaimen. Selain puolestaan käsittelee palautetun tiedon JavaScriptin avulla ja päivittää sivun tietoa käyttäjälle nähtäväksi. (W3Schools 2011a.)



KUVIO 13. AJAX-toiminnan periaate (W3Schools 2011)

Yhteenvedona AJAX:n eduista web-sovelluksia rakennettaessa voidaan todeta seuraavaa:

- Sen avulla voidaan rakentaa parempia ja responsiivisempia web-sivuja ja -sovelluksia.
- Sen suosion takia se rohkaisee mallien kehitykseen (patterns), tarkoittaen sitä, ettei kehittäjien tarvitse ”keksiä pyörää uudelleen” yleisimpien tehtävien suorittamisen osalta.
- Se hyödyntää jo olemassa olevia tekniikoita ja kehitystaitoja.
- AJAX-ominaisuudet integroituvat hyvin web-selaimien toiminnallisuuden kanssa.

AJAX:n mahdollisia ongelmakohtia ovat puolestaan:

- Käyttäjä ei voi lisätä AJAX-tuloksen sivua kirjainmerkkeihinsä, koska sivun osoite ei muutu (sivu ei lataudu uudelleen).
- Hakukoneet eivät välttämättä pysty indeksoimaan kaikkia AJAX-sivujen osioita.
- Selaimen takaisin-nappi ei tuota samaa tulosta kuin perinteisessä web-sivussa, koska kaikki tapahtuu saman sivun sisässä. Syy on sama kuin ensimmäisessä ongelmakohdassa.
- JavaScript voidaan laittaa selamiesta pois päältä, joka tekee AJAX-toiminnallisuuksista toimimattomia.

(Darie, Brinzarea, Chereches-Tosa & Bucica 2006, 16 - 17.)

3.5 MySQL Event Scheduler

3.5.1 Yleistä

MySQL Event Scheduler on ollut käytössä MySQL-versiosta 5.1.6. lähtien. Se mahdollistaa niin kutsuttujen tapahtumien (events) ajastamisen ja ajon. Nämä tapahtumat ovat tehtäviä, jotka suoritetaan määritellyn ajastuksen mukaisesti. Kun tapahtuma luodaan, niin tästä muodostuu tietokantaobjekti, joka sisältää suoritettavat SQL-lauseet. Suoritusten aikaväli voidaan määrittellä tapahtuvaksi tietyn ajanjakson väliajoin. Loppu- ja alkamisajankohdat on mahdollista niin ikään määrittellä tämän yhteydessä. Ideana tämä Event Scheduler muistuttaa UNIX-maailman crontabia ja Windows-puolella puolestaan Windows Task Scheduleria. (Oracle 2011.)

Tämän tyyppiset ajastetut tehtävät ovat kutsumanimeltään myös tunnettu *väliaikaisina laukaisijoina (temporal triggers)*, viitaten siihen, että nämä ovat objekteja, jotka laukaistaan ajan kuluessa. Näitä niin kutsuttuja laukaisijoita ei kuitenkaan pidä sekoittaa tapahtumiin. Laukaisija on tietokantaobjekti, jonka SQL-lauseet suoritetaan vastauksena määriteltyyn tapahtumaan, joka tapahtuu annetussa taulussa. Tapahtuma puolestaan on objekti, jonka SQL-lauseet suoritetaan tietyn ajankulumisjakson vastauksena. (Oracle 2011.)

MySQL-tapahtumilla on seuraavia keskeisiä ominaisuuksia:

- MySQL-versiosta 5.1.12 alkaen tapahtuma tunnistetaan yksilöllisesti nimen ja sille määritellyn mallin perusteella.
- Tapahtuma suorittaa tietyn toimenpiteen aikataulun mukaisesti. Tämä toimenpide koostuu SQL-lauseesta. Tapahtuman ajoitus voi olla joko kertaluontoinen tai toistuva. Kerran tapahtuva tapahtuma suoritetaan nimensä mukaisesti ainoastaan kerran. Toistuva tapahtuma puolestaan toistaa toimenpiteensä säännöllisen ajanjakson välein, ja ajastus tämän tyyppiseen tapahtumaan voidaan määrittää tietyn alkamis- ja (tai) loppumisajankohdan mukaisesti. Oletuksena toistuva tapahtuma alkaa heti, kun se on luotu, ja jatkuu loputtomasti, kunnes se poistetaan.

- Käyttäjät voivat luoda, muuttaa ja poistaa tapahtumia SQL-lauseiden perusteella. Nämä ominaisuudet sisältävät tapahtuman nimen, ajoituksen, persistenttisuuden, tilan (päällä/pois), suoritettavan toimenpiteen ja mallin, johon tapahtuma on määritelty. Tapahtumia muunnettaessa (alter) on huomioitava, että käyttäjällä tulee olla EVENT-oikeus tietokantaan, johon tapahtuma on luotu.

(Oracle 2011.)

Tapahtumat suoritetaan omassa, niille osoitetussa säikeessä (thread). Kun event scheduler on päällä, käyttäjät, joilla on PROCESS-oikeudet, voivat nähdä SHOW PROCESSLIST -komennolla tapahtumasäikeen ja sen tilan. Globaali järjestelmämuuttuja event_scheduler määrittelee, missä tilassa Event Scheduler on. MySQL-versiosta 5.1.12 lähtien tällä muuttujalla on kolme eri arvoa:

- OFF - event scheduler on pysäytetty, ja sen säie ei ole suorituksessa eikä ajastettuja tapahtumia suoriteta
- ON - event scheduler on käynnissä, ja sen säie ajaa ja suorittaa kaikki ajastetut tapahtumat
- DISABLED - event scheduler on toimimaton, sen säiettä ei ajeta, eikä sen tilaa voi muuttaa ajon aikana (ON/OFF).

(Oracle 2011.)

3.5.2 Toiminta

Kuviossa 14 on yksinkertainen esimerkkisyntaksi tapahtuman luonnista, jossa on annettu luonnin vähimmäisvaatimukset. Tämä luo tapahtuman nimeltä "myevent", joka suoritetaan kerran tunnissa, alkaen siitä kun tapahtuma on luotu (CURRENT_TIMESTAMP eli sen hetkinen aika). Tapahtuman nimen pitää olla validi MySQL-identifioija, ja siinä saa olla enintään 64 merkkiä. Tapahtuma assosioidaan skeemalla, ja jos sitä ei ole määritetty, käytetään oletuskeemaa. Tapahtumien nimet eivät myöskään ole kirjainkoosta riippuvaisia, joten ei voida määritellä esimerkiksi tapahtumia "myevent" ja "MyEvent" samaan skeemaan. Tapahtuma suorittaa SQL-lauseen, joka kasvattaa "mytable"-taulun "mycol"-kolumnin arvoa yhdellä. Taulun eteen on määritelty skeema "myschema".


```
CREATE EVENT myevent
  ON SCHEDULE AT CURRENT_TIMESTAMP + INTERVAL 1 HOUR
  DO
    UPDATE myschema.mytable SET mycol = mycol + 1;
```







































































































KUVIO 14. Esimerkki tapahtuman luonnista MySQL-syntaksilla (Oracle 2011)

4 TOTEUTUS

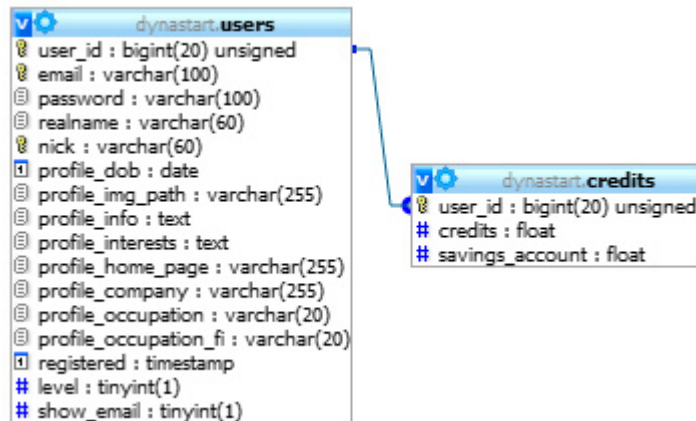
Toteutuksessa käydään läpi tehtyjä muutoksia. Painotus on teknisellä puolella, ja ulkonäköön ei juuri oteta kantaa.

4.1 Tietokantarakenne

Kuviossa 15 on tietokantataulut toteutukseen phpMyAdmin-ohjelmassa tarkasteltuna. Seuraavassa osiossa tutkitaan nykyisten taulujen muutoksia, ja sen jälkeen otetaan tarkasteluun uudet taulut. Projektien tunnistamiseen käytetään project_id-kenttää ja käyttäjät tunnistetaan user_id-kentän perusteella. Kuviossa 16 on esimerkkinä kuvattu users-taulun ja credits-taulun relaatio, jossa avaimena toimii user_id. Sivuston asennuksen yhteydessä categories-, faq-, static_strings-tauluihin lisätään ennakkoon asennustiedostossa määritellyjä tietoja ja luodaan yksi admin-oikeudet omaava käyttäjä (users-taulu).

	Table ▲	Action	Records ¹	Type	Collation
<input type="checkbox"/>	backed_projects	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	categories	     	13	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	credits	     	1	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	extra_credits_flag	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	faq	     	29	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	favourites	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	password_reset	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	projects	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	project_blog	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	project_comments	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	project_comment_replies	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	project_images	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	reports	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	site_blog	     	0	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	static_strings	     	271	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	users	     	1	MyISAM	utf8_unicode_ci
<input type="checkbox"/>	voted_projects	     	0	MyISAM	utf8_unicode_ci

KUVIO 15. Sivuston tietokantataulut



KUVIO 16. Esimerkki sivuston tietokantataulujen relaatioista

4.1.1 Muutokset nykyisiin tauluihin

Ensimmäisessä toteutuksessa olleet taulut olivat categories, favourites, projects, project_blog, project_comments, project_images, reports, site_blog ja users. Näistä tauluista muuttumattomina pysyvät favourites, project_blog, project_images ja reports. Categories-tauluun tuli uusina kenttinä category_en kategorian englanninkieliselle nimelle ja image_path kategorioita indikoivan kuvan tiedostonimelle. Projects-taulusta poistettiin ingress- ja project_goal-kentät. Video_path-kenttä korvattiin youtube_video_path- ja vimeo_video_path-kentillä, koska haluttiin mahdollisuus käyttää sekä YouTube- että Vimeo-videopalvelua. Uusina kenttinä tulivat updated kuvaamaan projektin päivytyspäivämäärää, thumbs_up ja thumbs_down projektin äänestysmäärille, promises projektin tekijän tai tekijöiden lupauksille, language projektin kielelle (toteutus tulevaisuudessa), visibility määrittelemään projektin näkyvyyttä sivulla sekä lopuksi sense- ja nuori_yrittäjyys-kentät määrittelemään, onko projekti osa SENSE-liikeideakilpailua ja Nuori yrittäjyys -ohjelmaa. Project_comments-taulun author-kenttä korvattiin user_id-kentällä, jolla tunnistetaan käyttäjän nimi tai nimimerkki users-taulusta. Kommentointi on mahdollista sekä nimellä että nimimerkillä, ja project_comments-tauluun lisättiin lisäksi kenttä identifier tunnistamaan kumpaa käyttäjä on käyttänyt. Site_blog-tauluun lisättiin user_id-kenttä, jonka perusteella voidaan tulostaa kirjoittajan nimimerkki blogimerkinnän yhteydessä.

Users-taulu koki muutoksia, jotka näkyvät lisääntyneinä kenttinä. Käyttäjä valitsee sivustolle rekisteröinnin yhteydessä itselleen uniikin nimimerkin, joka

sijoitetaan kenttään nick. Seuraavat lisätyt kentät liittyvät käyttäjän profiiliin, jota hän sivulle rekisteröidyttä voi päivittää. Tiedot ovat valinnaisia, ja niihin kuuluu syntymäaika, profiilikuva, tietoa käyttäjästä, kiinnostuksen kohteet, kotisivu, yritys (tai organisaatio, ryhmä), ammatti ja lopuksi toiminto, jos käyttäjä haluaa näyttää sähköpostiosoitteensa julkisesti muille (oletuksena ei näytetä, arvo 0). Kuviossa 17 on users-taulun rakenne phpMyAdmin-ohjelmasta.

Field	Type	Collation	Attributes	Null	Default	Extra
user_id	bigint(20)		UNSIGNED	No	None	AUTO_INCREMENT
email	varchar(100)	utf8_unicode_ci		No	None	
password	varchar(100)	utf8_unicode_ci		No	None	
realname	varchar(60)	utf8_unicode_ci		No	None	
nick	varchar(60)	utf8_unicode_ci		No	None	
profile_dob	date			Yes	NULL	
profile_img_path	varchar(255)	utf8_unicode_ci		No	None	
profile_info	text	utf8_unicode_ci		No	None	
profile_interests	text	utf8_unicode_ci		No	None	
profile_home_page	varchar(255)	utf8_unicode_ci		No	None	
profile_company	varchar(255)	utf8_unicode_ci		No	None	
profile_occupation	varchar(20)	utf8_unicode_ci		No	None	
profile_occupation_fi	varchar(20)	utf8_unicode_ci		No	None	
registered	timestamp			No	CURRENT_TIMESTAMP	
level	tinyint(1)			Yes	NULL	
show_email	tinyint(1)			No	0	

KUVIO 17. Users-taulun uusi rakenne

4.1.2 Uudet taulut

Toteutus sai aikaan myös tarpeen uusille tauluille. Backed_projects-taulu sisältää tiedon projekteista, joita käyttäjä on tukenut. Projekti tunnustetaan project_id- ja käyttäjä user_id-kentän perusteella. Taulusta löytyvät myös kentät sijoitetulle krediittimäärälle ja lippukenttä kuvaamaan, onko käyttäjältä veloitettu epäonnistuneesta projektista hänen sille antama krediittimäärä. Credits-taulu sisältää user_id-, credits- ja savings_account-kentät. Credits-kenttä sisältää käyttäjän käytettävissä olevat krediitit ja savings_account krediittisumman, joka on sijoitettu voimassa oleviin tai onnistuneisiin projekteihin. Extra_credits_flag-taulun ainoaa extra_credits_paid_flag-kenttää käytetään tunnistamaan päivän ensimmäinen sisäänkirjautuminen, joka palkitaan lisäkrediiteillä.

Usein kysytyt kysymykset (FAQ) -sivua varten luodaan faq-taulu. Kenttien tiedot sisältävät muun muassa kunkin osion identifioijan, otsikot ja aliotsikot, sekä

sisällön. Jokaisesta näistä kentästä on kaksi versiota, toinen suomen kielelle ja toinen englanniksi. Salasanan unohtamista varten on puolestaan password_reset-taulu. Se sisältää käyttäjän sähköpostiosoitteen sekä selkokielisenä että md5-salattuna ja lisäksi md5-salauksen myös sen hetkisestä session_id:stä. Näitä kahta salausta käytetään linkin muodostukseen salasanaa vaihdettaessa. Expires-kenttään tulee aikaleimatieto linkin vanhenemisesta.

Projektin kommenttien vastaamiseen kirjoitettu tieto lisätään project_comment_replies-tauluun. Tunnistamiseen käytetään comment_id-kenttää alkuperäiselle kommentille, project_id:tä projektiin, johon kommentti kuuluu ja user_id:tä käyttäjän määrittämiseen. Kommentoinnin tavoin, myös kommenttivastauksissa on mahdollisuus käyttää sekä nimeä tai nimimerkkiä, joten tästä taulusta löytyy myös kommenttitaulun tavoin identifier-kenttä. Kuviossa 18 on kommenttivastaustaulun rakenne kokonaisuudessaan phpMyAdmin-näkymässä.

Field	Type	Collation	Attributes	Null	Default	Extra
reply_id	bigint(20)		UNSIGNED	No	None	AUTO_INCREMENT
comment_id	bigint(20)		UNSIGNED	No	None	
project_id	bigint(20)		UNSIGNED	No	None	
user_id	bigint(20)		UNSIGNED	No	None	
identifier	varchar(30)	utf8_unicode_ci		No	None	
created	timestamp			No	CURRENT_TIMESTAMP	
content	text	utf8_unicode_ci		No	None	

KUVIO 18. Project_comment_replies-taulu

Static_strings-taulu sisältää sivuston suomen- ja englanninkieliset merkkijonot. Jokaiselle sanalle tai lausekokonaisuudelle on tunnistuksen osalta uniikki identifier-kenttä, page-kenttä, joka kertoo, missä sivuston sivulla tai sivuilla kyseistä merkkijonoa käytetään, sekä value- ja value_en-kentät itse merkkijonoille. Viimeinen voted_projects-taulu sisältää tiedon projekteista, joita käyttäjä on joko ylä- tai alapeukuttanut. Yksittäistä projektia voi peukuttaa vain kerran, joten tällä taululla saadaan selville, mitä projekteja käyttäjä on jo äänestänyt (thumbs_up- ja thumbs_down-kentät projects-taulussa), ja näin rajataan nämä projektit pois. Esimerkiksi backed_projects-taulun tavoin, käyttäjä tunnistetaan user_id-kentän ja projektia project_id-kentän perusteella.

4.2 Sivutus

Kun tietoa esintyy yhdessä sivussa runsaasti, on vaihtoehtona käyttää sivutusta, jossa tulostetaan tietokannasta aina tietty määrä tietoa. Sivusto käyttää sivutusta seuraavissa osioissa:

- projektien kategoriaselaus
- projektien blogit
- projektien kommentit
- projektien tukijat
- projektien hakutulokset
- käyttäjän suosikkiprojektit
- käyttäjän omat projektit
- käyttäjän tukemat projektit
- sivustoblogi
- admin-hallintapaneeli.

Rivejä tulostetaan tietokannasta tietty määrä osiosta riippuen. Esimerkiksi projektin kommentteissa yhden sivulle tulee 8 kommenttia, ja rajoitus toteutetaan MySQL:n LIMIT-komennolla. Sivunumerot tulostetaan aina vasta siinä tapauksessa, kun sivuja on 2 tai enemmän. Sivutus on toteutettu MySQL:n, PHP:n ja jQuery:n ajax-metodin kombinaatiolla. Ensimmäistä kertaa osiota ladattaessa sivun HTML muodostuu PHP:n ja MySQL:n avulla, mutta sivunumeroita sekä eteen- ja taaksepäin painettaessa haetaan AJAX:lla pager.php:sta tietoa POST-metodia käyttäen.

Esimerkkinä on projektien kategoriaselausosio, jossa projektien määrä sivua kohden on rajoitettu 6:n. Sivunumeroiden tulostuksen logiikka on sama kaikissa osioissa. Kuviossa 19 on kategoriaselauksen sivunumeroiden tulostus. Rivimäärä saadaan MySQL-kyselystä, jossa `mysql_num_rows`-funktio palauttaa ne rivit, jonka kategoria-id on GET-metodin `cat`-parametrin arvo. Kysyttäessä URL-osoitteessa kulkevien parametrien arvoja MySQL:ssä on tarpeellista käyttää PHP:n `mysql_real_escape_string`-funktioita estämään SQL-injektiot, eli `$category_id`-muuttujan arvo on `mysql_real_escape_string($_GET['cat'])`. Lisäksi halutaan rivimäärään mukaan vain ne projektit, jotka ovat julkisesti näkyvillä

(visibility-arvo 1). Ceil-funktio palauttaa seuraavan suurimman integer-tyyppisen arvon, eli pyöristää sen, jos tarve vaatii. Jos tuo arvo on enemmän kuin 1, tulostetaan sivunumerot ja eteen- sekä taaksepäin-napit.

```

81     $per_page = 6;
82
83     $count_query = mysql_query("SELECT project_id FROM projects
84     WHERE category_id=".$category_id." AND visibility=1");
85     $count = mysql_num_rows($count_query);
86     $pages = ceil($count/$per_page);
87
88     if ($pages > 1)
89     {
90         echo "<div id='browse_page_numbers'>";
91         echo "<ul id='browse_numbers'>";
92         //Pagination Numbers
93         echo "<li><a id='back_button'>&lt;&lt;</a></li>";
94         for($z=1; $z<=$pages; $z++)
95         {
96             echo "<li><a id='".$z."' class='p_numbers'>".$z."</a></li>";
97         }
98         echo "<li><a id='next_button'>&gt;&gt;</a></li>";
99         echo "</ul>";
100        echo "</div>";
101    }

```

KUVIO 19. Sivunumeroiden tulostus

Kuviossa 20 on kategoriaselausfunktion toteutus, joka ottaa parametrinä sivunumeron. Katogoria-id ja haluttu sivutussivu (switch-case -rakenne pager.php:ssa) ovat hidden-tyyppisiä input-kenttiä. Pager.php:ssa on toteutus sekä suomen- että englanninkielisellä kategoriasivutusselaukselle, jotta URL-osoite muodostuu projektia painettaessa oikein. Pager.php:ssa tehtävissä kyselyissä rajoitetaan tuloksia antamalla LIMIT-komennolle alkuarvo (sivunumero - 1) * rivejä per sivu sekä loppuarvo rivejä per sivu, eli esimerkiksi sivunumeroa 2 painettaessa arvot ovat 6,6. Ensimmäinen arvo on ensimmäinen rivi joka palautetaan (alkaa 0:sta), ja toinen arvo on palautettavien rivien määrä. Latausvaiheessa project_items-diviin näytetään latauskuva siihen asti kunnes ajax-funktio saavuttaa success-tilan, ja tulostetaan tieto pager.php:sta (msg).

Kuviossa 21 on loadBrowseData-funktion käyttö sivunumeroihin (eteen- ja taaksepäinnapeille on oma logiikka sivunumeron määrittämisessä, mutta käytettävä loadBrowseData-funktio pysyy samana). Painettavaan numeroon lisätään active_number-luokka, jonka CSS-määrittäykseen kuuluu alleviivattu teksti, sekä poistetaan se muilta (oletuksena sivunumero 1 on alleviivattu)

sivunumeroelementeiltä. LoadBrowseData-funktion sivunumero-parametri saadaan kunkin sivunumero-elementin id-arvolla. Kuviossa 22 on käyttäjälle näkyvä tilanne. Käyttäjä on sivunumerolla 2, jossa näkyy 7. projekti kyseisestä kategoriasta.

```

24 // projektien selaus
25 function loadBrowseData(page)
26 {
27     |
28     |
29     |   var catId = $("#cat_id").val();
30     |   var pager = $("#browse_lang").val();
31     |
32     |   // ie fix
33     |   $("#project_items").css("-ms-filter", "progid: DXImageTransform.Microsoft.Alpha(Opacity=20)");
34     |   $("#project_items").html("<img src='img/ajax-loader.gif' />");
35     |
36     |
37     |   $.ajax
38     |   (
39     |     type: "POST",
40     |     url: "content/pager/pager.php",
41     |     data: "cat_id="+catId+"&p="+page+"&pager="+pager,
42     |     success: function(msg)
43     |     {
44     |       $("#project_items").ajaxComplete(function(event, request, settings)
45     |       {
46     |         $("#project_items").html(msg);
47     |       });
48     |     }
49     |   );
50 }

```

KUVIO 20. Kategoriaselausfunktion toteutus

```

369 /* projektien selaus */
370 $("#browse_page_numbers ul#browse_numbers li .p_numbers").click(function() {
371     |
372     |   $("#browse_page_numbers ul li .p_numbers").removeClass("active_number");
373     |   $(this).addClass("active_number");
374     |
375     |   //Loading Data
376     |   var pageNum = this.id;
377     |   loadBrowseData (pageNum);
378     |
379     |
380     | });

```

KUVIO 21. Kategoriaselausfunktion käyttö sivunumero-elementteihin



KUVIO 22. Kirjallisuus-kategorian projektien selaus

4.3 Ennakoiva haku

Ennakoiva haku, joka on jQuery UI:n autocomplete-widget, hakee sivustolta projekteja niiden otsikon perusteella. Toiminnon myötä käyttäjän ei tarvitse kirjoittaa kuin projektin otsikon alkuosa, minkä seurauksena hakukentän alle ilmestyy ehdotuksia projekteista, joita haetaan.

Ennakoivan haun tulokset eroavat sivustolla perinteisestä hausta tietokantakyselyn osalta. Perinteisessä haussa tuloksia haetaan hakuun syötetyn merkkijonon perusteella sanan ympäriltä käyttämällä LIKE-komentoa sekä prosenttimerkkejä (%) haettavan merkkijonon molemmin puolin. Sen lisäksi projekteja haetaan myös niille annettujen avainsanojen perusteella, johon sovelletaan niin ikään LIKE-komennolla molemmin puolin olevaa esiintyvyyttä, jotka saadaan aikaan prosenttimerkeillä tietokantakyselyssä. Ennakoivassa haussa haetaan projektin nimeä tietokannasta LIKE-komentoa käyttäen, mutta prosenttimerkki on vain termin oikealla puolella, joka tarkoittaa hakutuloksien kannalta sitä, että haettavien projektien nimet pitävät alkaa samoilla merkeillä kuin käyttäjän syöte.

Kuviossa 23 on ennakoivan haun jQuery UI -koodiosuus. Tekstihakukentän id on searchbox, johon autocomplete-metodia käytetään. Lähteenä on erillinen php-tiedosto, jossa suoritetaan tietokantahaku GET-parametrille term. Kuviossa 24 on autocomplete.php:ssa tälle term-parametrille tehtävä tietokantakysely, joka on rajoitettu 6:n tulokseen luontipäivän mukaan. Title-kenttää tarvitaan projektin nimen muodostamiseen ja project_id-kenttää puolestaan URL-osoitteeseen, johon käyttäjä projektin valitessaan ohjautuu. Projektien näkyvyys pitää olla julkinen, sekä niiden aikajakson pitää olla voimassa. Deadline-kenttää verrataan \$today-muuttujaan, johon saadaan nykyinen aika time-funktiolla, sekä date-funktiolla se oikeaan formaattiin. Kyselystä saadut tulokset käydään läpi while-silmukassa, jossa ne lisätään taulukkomuuttujaan array_push-funktiolla. Lopuksi silmukan jälkeen tulostetaan taulukkomuuttujan tieto json-muodossa json_encode-funktiolla.

Ennakoiva haku aktivoituu, kun käyttäjä on syöttänyt tekstikenttään vähintään 2 merkkiä. Tuloksia voidaan selata nuolinäppäimillä ylös ja alas, ja ne voidaan valita enterillä tai hiirenpainalluksella, jolloin käyttäjä ohjautuu valitulle projektisivulle (englannin- ja suomenkielelle on erilliset osoitteet). Valitessa tekstihakukentän disabled-attribuutille annetaan arvo "disabled", sillä muuten se aiheuttaa ongelmia Mozilla Firefox -selaimessa. Tämä ongelma on se, ettei käyttäjää ohjatakaan halutulle projektisivulle, vaan tehdään perinteinen haku, jonka tulokset näytetään erillisellä sivulla.

Kuviossa 25 on esimerkki ennakoivasta hausta. Käyttäjä on syöttänyt hakukenttään "lo", ja vienyt hiiren "Lorem ipsum 2" -nimisen projektin päälle. Jos projekteja liikutellaan ylös- ja alas-nuolinäppäimillä, päivittyy myös hakukentän merkkijono valitulla olevalle projektille. Koska prosenttimerkki on tietokantakyselyssä vain termin oikealla puolella, ei Lorem ipsum 1-6 -projektien tuloksia näy ennakoivassa haussa esimerkiksi kirjoittamalla hakukenttään "ip". Huomioitavaa on myös, ettei kirjasinkoolla ole hakua tehdessä merkitystä. Jos kursorin kohdistus on searchbox-tekstikentässä ja käyttäjä painaa enteriä tai klikkaa hiirellä etsi-painiketta, suoritetaan perinteinen haku.

```

1316 // autocomplete
1317 $("#searchbox").autocomplete({
1318     source: "sql/autocomplete.php",
1319     minLength: 2,
1320     select: function( event, ui ) {
1321         $('#searchbox').attr("disabled","disabled"); // firefox fix
1322
1323         if ($("#searchbutton").val() == "etsi")
1324             window.location = "?page=project&pid="+ui.item.pid;
1325         else if ($("#searchbutton").val() == "search")
1326             window.location = "?page=project&pid="+ui.item.pid+"&lang=en";
1327     }
1328 });

```

KUVIO 23. JQuery UI:n autocomplete-metodi

```

9 $result = mysql_query("SELECT project_id, title FROM projects
10 WHERE visibility=1 AND deadline > '". $today.'" AND
11 title LIKE '". $term.'" ORDER BY added LIMIT 6");

```

KUVIO 24. Ennakoivan haun tietokantakysely



KUVIO 25. Ennakoivan haun toiminta sivustolla

4.4 Käyttäjätili

Rekisteröitymällä sivustolle saa käyttöönsä lisää toiminnallisuutta.

Rekisteröintilomakkeessa kysytään pakollisina tietoina nimi, -merkki, sähköpostiosoite ja salasana. JQueryn puolella tehdään asiakaspuolen kenttien tarkistusta muuan muassa nimimerkin sallituista merkeistä sekä kelvollisesta sähköpostiosoitteesta. Kirjautunut käyttäjä voi

- luoda, muokata ja poistaa projektin
- kirjoittaa projektikohtaisia blogeja
- kommentoida projekteja, ja vastata kommentteihin
- lisätä suosikkeihin muiden projekteja

- tukea ja peukuttaa muiden projekteja
- selata tukemishistoriaansa
- muokata käyttäjätilitietojaan ja poistaa käyttäjätilin.

Kuviossa 26 on oman käyttäjätilin aloitusnäkökuva. Lähes identtinen näkökuva on myös muiden käyttäjien tietoja tarkasteltaessa, joihin pääsee kommenttien kautta painamalla kommentoijan nimeä tai nimimerkkiä, joka on a-elementti. Muiden käyttäjien tiedot haetaan URL-osoitteessa kulkevan GET-parametrin ”nick”-arvon perusteella, ja oman profiilin tiedot puolestaan kirjautumisen yhteydessä session-muuttujaan sijoitetun user_id:n perusteella.



The image shows a user profile interface. On the left, there is a placeholder for a profile picture and the user's name 'Nimi: Teppo Testaaja' and nickname 'Nimimerkki: Teppo123'. On the right, there is a table of user information with the following fields:

Sähköpostiosoite:	teppo@testaaja.com
Syntymäaika:	08.03.1982 (ikä 29)
Tietoa minusta:	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed posuere interdum sem.
Kiinnostuksen kohteet:	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed posuere interdum sem.
Kotisivut:	http://google.fi
Yritys / organisaatio / ryhmä:	Lorem ipsum Oy
Ammatti:	opiskelija

At the bottom right of the table, there are three buttons: 'Muokkaa profiilia', 'Poista käyttäjätili', and 'Takaisin'.

KUVIO 26. Käyttäjätilin hallinta

Suosikkeihin lisääminen, projektien tukeminen ja peukutus toteutetaan JQuery:n ajax-metodilla, jossa lähteenä on erillinen php-tiedosto. Projektisivu sisältää hidden-tyyppisen input-kentän, jonka arvona on projektin id, jotta tiedetään, mitä projektia tuetaan ja peukutetaan. Jokaista projektia saa peukuttaa (ylös tai alas) ja tukea vain kerran.

Kuviossa 27 on projektisivun tukemis- ja peukutusosuosuus. Käyttäjä on painanut yläpeukutusta, jolloin projects-tilin thumbs_up-kentän arvoa kasvatetaan yhdellä. Kun php-skripti on ajettu ja ajax-metodi saavuttaa success-tilan, kasvatetaan ylöspeukutusmäärää yhdellä sekä annetaan vihreälle palkille oikea leveysprosentti. Nämä HTML:n muokkaukset tapahtuvat ilman sivunlataamista,

ja se toteutetaan jQuery html-metodilla. Kun käyttäjä tulee ensimmäisen kerran kyseiselle projeksisivulle, luetaan tiedot tietokannasta ja näytetään ne PHP:lla dynaamisesti ilman jQueryn puuttumista asiaan. Projektin tukeminen noudattaa samaa periaatetta kuin projektin peukutus.



KUVIO 27. Projektien tukeminen ja peukutus

4.4.1 Krediittijärjestelmä

Käyttäjällä on käytettävänä 1000 krediittiä kuukautta kohden. Hän voi sijoittaa muiden projekteihin minkä tahansa kokonaislukuprosentin 1 ja 1000 krediitin väliltä, kunhan tilillä on tarpeeksi krediittejä. Sijoitettu summa vähennetään käyttäjän tililtä sekä lisätään erilliselle sijoitustilille. Jos projekti epäonnistuu tavoitteessaan (kerätty summa on vähemmän kuin tavoite päivien loputtua), vähennetään sijoitustililtä käyttäjän sille sijoitettu summa. Tämä suoritetaan

sivustolle kirjautumisen yhteydessä signin.php:ssa, jossa tehdään kysely projekteista, jotka eivät ole saavuttaneet tavoitettaan aikarajaan mennessä. Jos käyttäjä on tukenut näitä, vähennetään credits-taulun savings_account-kentästä kyseinen summa ja merkitään backed_projects-taulun money_taken_flag-kentän arvoksi 1, jottei erotusta enää tehdä uusia kertoja.

Credits-taulun credits-kenttä päivitetään joka kuukauden ensimmäinen päivä kello 00:00:00 arvoksi 1000 käyttäen MySQL Event Scheduleria. Toinen MySQL Event Schedulerilla tehtävä toiminto on päivän ensimmäisen kirjautuneen käyttäjän palkitseminen 10 ekstrakrediitillä, joka toteutetaan vaihtamalla extra_credits_flag-taulun extra_credits_paid_flag-kentän arvoksi 0 joka päivä kello 00:00:00. Kirjautumisen yhteydessä katsotaan, mikä tämän kentän arvo on. Jos se on 0, niin kasvatetaan kyseisen käyttäjän krediittejä 10:llä ja päivitetään extra_credits_paid_flag-kentän arvoksi 1. Tämä toteutus on kuvattu kuviossa 28. Kuviossa 29 on näiden kahden tapahtuman toteutus phpMyAdmin-ohjelmassa tarkasteltuna. Tietokannan tapahtumat saa näkyviin MySQL-komennolla ”SHOW EVENTS”.

```

47 // 10 ekstrakrediittiä päivän ensimmäiselle kirjautujalle
48 $fQuery = mysql_query("SELECT extra_credits_paid_flag
49     FROM extra_credits_flag");
50 $fRow = mysql_fetch_assoc($fQuery);
51
52 if($fRow['extra_credits_paid_flag'] == 0)
53 {
54     mysql_query("UPDATE credits SET credits=credits+10
55     WHERE user_id=". $_SESSION['dynafund']['id']);
56     mysql_query("UPDATE extra_credits_flag
57     SET extra_credits_paid_flag=1");
58 }

```

KUVIO 28. 10 lisäkrediitin toteutus

Db	Name	Definer	Time zone	Type	Execute at	Interval value	Interval field	Starts	Ends	Status
dynastart	creditsupdate	root@localhost	SYSTEM	RECURRING	NULL	1	MONTH	2011-11-01 00:00:00	NULL	ENABLED
dynastart	extra_credits	root@localhost	SYSTEM	RECURRING	NULL	1	DAY	2011-10-18 00:00:00	NULL	ENABLED

KUVIO 29. MySQL Event Scheduler tapahtumat dynastart-tietokannassa

4.4.2 Muokkaus

Käyttäjätietojaan pääsee muokkaamaan painamalla Muokkaa profiilia -nappia. Käyttäjä ohjataan erilliselle muokkaussivulle, jossa on mahdollista päivittää pakollisia tietoja, valinnaisia tietoja sekä vaihtaa salasana. Kenttien arvoiksi tulostetaan aina sen hetkiset tiedot tietokannasta, jotta niitä ei tarvitse kirjoittaa ja muuttaa tyhjästä. Kyseessä on lomake, joka lähetetään prosessoitavaksi erilliselle php-tiedostolle, ja jonka jälkeen käyttäjä ohjataan käyttäjätilihallinnan aloitusnäkyeseen. Koska käyttäjän on mahdollista ladata profiilikuva itselleen, täytyy form-elementin enc-type -attribuutin arvon olla ” multipart/form-data”.

Pakollisten kenttien osalta on tarpeen katsoa jQuerylla, ettei esimerkiksi nimi-kenttä ole tyhjä. Jos käyttäjä yrittää päivittää tietojansa pakollisen kentän ollessa virheellinen, lomakkeen lähetys estetään preventDefault-funktiolla ja määritellään CSS:llä virheellisten kenttien reunuksien väriksi punainen. Lisäksi katsotaan nimimerkki-kentän näppäinpainalluksia, ja kentän arvoa tietokannasta, onko kyseinen nimimerkki jo käytössä. Kuviossa 30 on kuvattu tilanne, jossa käyttäjä on yrittänyt lähettää lomakkeen virheellisillä tiedoilla.

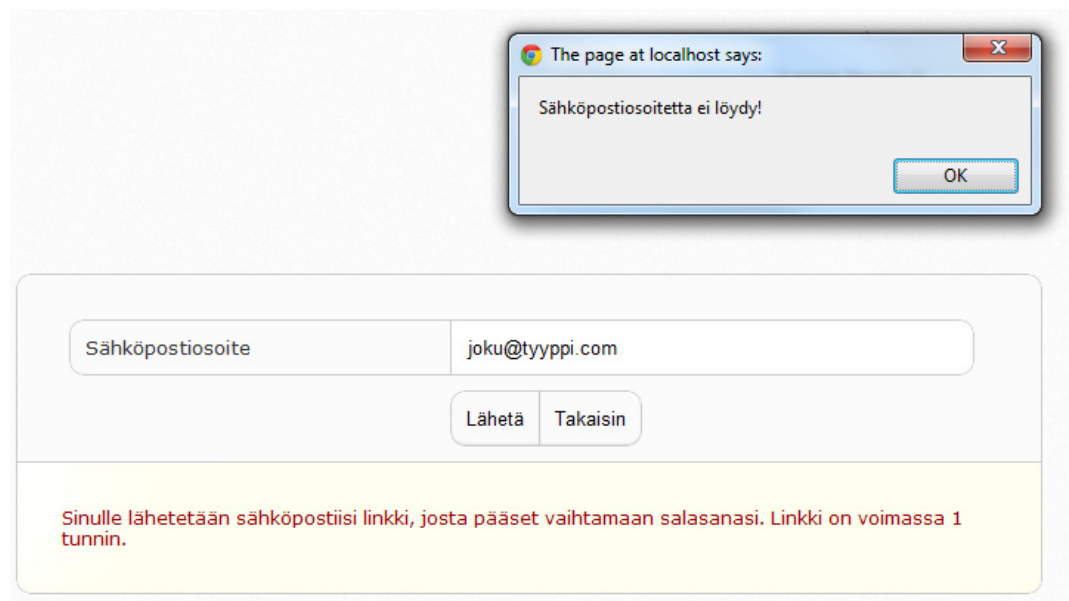
The screenshot shows a web form titled "Perustiedot" (Basic information). It contains several input fields and a file upload button. The "Nimimerkki" (Username) field is highlighted with a red border and a red message box below it that says "Nimimerkki käytössä!". The "Etu- ja sukunimi" (First and last name) field is also highlighted with a red border. The "Sähköpostiosoite" (Email address) field contains the value "teppo@testaaja.com". The "Valitse uusi profiilikuva" (Choose new profile picture) field has a "Choose File" button and the text "No file chosen". At the bottom, there is a checkbox labeled "Näytä sähköpostiosoitteeni muille" (Show my email address to others).

KUVIO 30. Virhetilanne käyttäjäprofiilitietojen päivityksessä

4.4.3 Salasana unohtunut

Salasanan vaihtamiseen kuuluu kolme eri vaihetta: sähköpostiosoitteen täyttäminen lomakkeeseen, sähköpostiin tulevan linkin painaminen sekä uuden salasanan asettaminen lomakkeella.

Sivulla, jossa sähköpostiosoite annetaan, katsotaan ensin, ettei käyttäjä ole kirjautunut, ja katsotaan myös ettei PHP:n session_id:n arvo ole tyhjä (jos on, niin aloitetaan istunto PHP:n session_start-funktiolla). Lomaketta ei lähetetä perinteisesti (preventDefault), vaan käytetään jQueryn ajax-metodia erillisellä PHP-skriptillä. Jotta tämä skripti saa session tiedot, on siellä kutsuttava myös session_startia, jolla voidaan myös jatkaa istuntoa. Jos tietokannasta ei löydy käyttäjän syöttämää sähköpostiosoitetta, näytetään siitä JavaScriptin alert-ikkunalla tieto. Kuviossa 31 on kuvattu tilanne, jossa käyttäjä on antanut sähköpostiosoitteen, jota ei löydy tietokannasta.



KUVIO 31. Käyttäjä on antanut sähköpostiosoitteen, jota ei löydy tietokannasta

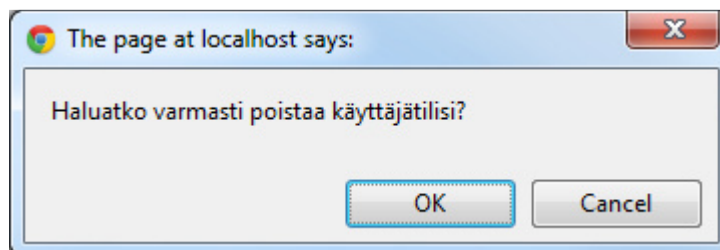
Kun käyttäjä antaa sähköpostiosoitteen, joka löytyy tietokannasta, lisätään ajax-metodin lähdetiedostossa password_reset-aulun email-kenttään kyseinen sähköpostiosoite, email_hash-kenttään kyseinen sähköpostiosoite md5-salattuna, sid_hash-kenttään session id md5-salattuna ja expires-kenttään aika tunnin päästä. Tämä aika saadan käyttämällä PHP:n time-funktiota, jonka arvoon lisätään 3600

(60 min. * 60 sec.). Lisäksi lähetetään käyttäjän antamaan sähköpostiosoitteeseen linkki, josta käyttäjä pääsee asettamaan uuden salasanan. Lisäksi kutsutaan session_regenerate_id-funktiota, jotta saadaan jokaisesta taulun rivistä uniikimpi. Käyttäjän painettua alert-ikkunan OK-nappia hänet uudelleenohjataan sivuston etusivulle.

Sähköpostiosoitteeseen lähetetty linkki muodostuu GET-parametrien osalta URL-osoitteessa mukana liikkuvista parametreista "email" ja "sid", jotka ovat molemmat md5-muodossa. Salasanavaihtosivulla tarkastellaan password_reset-taulun email_hash-kentän, sid_hash-kentän ja expires-kentän arvoja. Expires-kentän arvoa verrataan nykyiseen aikaan, joka saadaan time-funktiolla. Jos email- ja sid-parametrien arvot URL-osoitteessa täsmäävät tietokannasta löytyviin, ja sähköpostiin tullutta linkkiä painetaan tuntiin mennessä sen lähettämistä, näytetään käyttäjälle lomake, johon asetetaan uusi salasana. Varmistukseksi salasana kirjoitetaan kaksi kertaa. Lomake on jälleen ajax-metodilla toteutettu, ja jQuerylla katsotaan lisäksi, että annetut salasanakentät täsmäävät ja täyttävät sille annetut kriteerit. Onnistuneen vaihdon jälkeen näytetään alert-ikkuna, jossa kerrotaan salasanan vaihdosta, sekä ohjataan käyttäjä etusivulle.

4.4.4 Poisto

Käyttäjän on mahdollista poistaa tilinsä sivustolta. Nappi tälle toiminnolle löytyy käyttäjätilin hallinnasta. Painettaessa Poista käyttäjätili -nappia käyttäjältä kysytään JavaScriptin confirm-metodia käyttäen, haluaako käyttäjä varmasti poistaa tilinsä, joka on kuvattu kuviossa 32.



KUVIO 32. Käyttäjätilin poistovarmistus

Poistettaessa käyttäjätili poistetaan tietokannasta kaikki kyseistä käyttäjää koskevat tiedot, poislukien ne, jotka vaikuttavat muiden projekteihin. On myös huomioitavaa, että kaikki käyttäjän tekemät projektit poistetaan myös. Projektin kuvien osalta poistetaan sekä tietokantaviittaukset että fyysiset kuvat. Fyysiset kuvat poistetaan PHP:lla unlink-funktiolla, jonka käyttö varmistetaan is_file-funktiolla, jolla katsotaan, onko kyseisiä tiedostonimiä olemassa.

Poistotoimenpiteiden jälkeen käyttäjä saa alert-viestin, jossa kerrotaan onnistuneesta tilin poistosta, ja hänet ohjataan logout.php-sivulle, jossa tuhoetaan istunto ja alustetaan istuntomuuttujataulukko.

4.5 Projektit

Projektit ovat edelleen sivuston idean lähtökohta. Luontivaiheessa täytettäviin kenttiin on tullut muutoksia sekä käyttäjälle on lisäksi annettu mahdollisuus muokata lennosta projektejaan esimerkiksi virheellisten näppäinpainallusten korjaamista silmälläpitäen. Uutena ominaisuutena on myös projektin poistotoiminto. Projektikuvien selaamiseen käytetään Fancybox-nimistä jQueryn lisäosaa, joka avaa kuvat selattaviksi erilliseen dialogiin. Projektin näkyvyys voi olla joko 0 tai 1 (visibility-kentän arvo projects-taulusta), ja sitä hallinnoi admin-oikeudet omaava käyttäjä hallintapaneelista käsin. Jos näkyvyys on 0, käyttäjä näkee projektinsa vain omat projektit -sivulla, ja jos näkyvyys on 1, niin projekti näytetään julkisesti koko sivulla.

4.5.1 Luonti

Projektin luonti sisältää sekä pakollisia että valinnaisia kenttiä, jotka täytetään ja valitaan projektinluontilomakkeella. Pakollisiin kenttiin kuuluvat projektin nimi, kategoria, lyhyt kuvaus, pitkä kuvaus, avainsanat, tavoitekrediittimäärä, videoupotus, kesto, yritys, toiminta ja yhteystiedot. Valinnaisia kenttiä ovat puolestaan kuvat, SENSE-liikeideakilpailuun osallistuminen, Nuori yrittäjyys-toimintaan kuuluminen, kotisivut ja lupaukset.

Lomake lähetetään prosessoitavaksi erilliselle php-tiedostolle, jossa lisätään annettuja tietoja tietokantaan, prosessoidaan mahdolliset projektikuvat

(alkuperäisestä kuvista tehdään kaksi kokoversiota, alkuperäiset kuvat poistetaan) ja lopuksi ohjataan käyttäjä tehdyille projektisivulle. Käyttäjätilitietojen päivityksen tavoin, lomakkeen enctype-attribuutin arvon pitää tiedostokenttien takia olla ” multipart/form-data”.

Osa täytettävistä kentistä sisältää myös edellisessä toteutuksessa käytössä olleen jQuery UI dialog -toiminnon, jolla näytetään käyttäjälle erillisessä ikkunassa lisäopastusta kentän täyttämiseen. Jokaiselle näistä dialogeista on annettu määrityksenä autoOpen arvo ”false”, width ”415”, ja resizable ”false”. Tämä tarkoittaa, että dialogi ei avaudu sivulatauksen yhteydessä, niiden leveys on 415 pikseliä ja niitä ei voi suurentaa. Jokaiselle dialogisisällölle, joka on div-elementin sisällä, on annettu luokka ”helpDialog”. Painamalla kenttien lopussa olevaa kysymysmerkkiä avautuu kenttäkohtainen dialogi-ikkuna.

Lisäopastuksien toteutus on havainnollistettu kuviossa 33, jossa on esimerkkinä videoapudialogin avaaminen. VideoQ-id:n omaavaa a-elementtiä painettaessa estetään ensin selaimen osoitteen muutos preventDefault-funktiolla, minkä jälkeen avataan dialogi. Kuviossa 34 on nähtävissä kyseinen videoapudialogi, joka käyttäjälle näkyy. Dialogia voi oletuksena liikutella hiirellä.

```

933 // Dialog
934 $('#helpDialog').dialog({
935     autoOpen: false,
936     width: 415,
937     resizable: false
938 });
939
940
941 $('#videoQ').click(function(e) {
942     e.preventDefault();
943     $('#videoDialog').dialog('open');
944     return false;
945 });

```

KUVIO 33. Dialog-widgetin ominaisuuksien määrittely ja käyttö

Choose File No file chosen

Video YouTubesta tai Vimeosta * [?]

Valitse videotarjoaja: YouTube

http://www.youtube.com/watch?v=

Kesto * [?]

Videon lisääminen ✕

YouTube
 Syötä merkit jotka tulevat ensimmäisen "=" -merkin jälkeen.
 Esimerkiksi: www.youtube.com/watch?v=sBWPCvdv8Bk.
 Oikea syöte: **sBWPCvdv8Bk**

Vimeo
 Syötä numerot jotka tulevat ensimmäisen kauttaviivan ("/") jälkeen.
 Esimerkiksi: www.vimeo.com/21116038.
 Oikea syöte: **21116038**

KUVIO 34. Videoapu-dialogi

Projektin luontilomake sisältää myös kentille tehtäviä tarkistuksia. Esimerkiksi tavoitekrediittimäärän minimimäärä on 1000 ja maksimimäärä on 50000. Lisäksi näihin krediittikenttiin voi syöttää näppäimistöltä ainoastaan numeroita.

Krediittikenttien numeroarvo päivittyy kokonaiskrediittimäärään. Krediittikenttiä voidaan lisätä ”Lisää uusi kenttä” -napilla, joka käyttää lisäämiseen jQueryn appendTo-metodia. Kuviossa 35 on kyseisen toiminnan näkymä.

Kuinka paljon virtuaalirahaa haluaisit saada? * [?]

Yhteisminimimäärä on 1000, ja -maksimimäärä on 50000. Oikea syöttömuoto: 1250, 1500, 2225, ..

3000	Tilat
400	Tarvikkeet
Määrä	Käyttötarkoitus

Lisää uusi kenttä

Joka tekee yhteensä 3400 D

KUVIO 35. Projektiluonnin krediittimääräkentät

4.5.2 Muokkaus

Omaa projektia on mahdollista muokata projektisivulla. Projektin luontivaiheessa projects-taulun host_id-kenttään sijoitetaan käyttäjän user_id. Tämä kyseinen user_id-arvo löytyy myös session-taulukkomuutujasta. Jos nämä arvot täsmäävät, tulostetaan projektisivulle myös lomake, joka on oletuksena piilossa. ”Muokkaa projektia” -nappia painettuaan sivua muutetaan piilottamalla tiedot jQueryn hide-

metodilla ja näyttämällä kentät show-metodilla. Muokkaa-nappi piilotetaan, ja näytetään ”Peruuta muutokset” ja ”Tallenna muutokset” -napit. Kenttien arvot ovat valmiiksi samoja, sillä ne luetaan myös tietokannasta. Nykyisillä kentillä ja muokattavilla kentillä on luokka-attribuutteja, jolloin niiden piilottaminen ja näyttäminen onnistuu jouhevasti.

Projektista on mahdollista muuttaa nimeä, lyhyttä kuvausta, pitkää kuvausta, videoupotetta, SENSE-liikeideakilpailuun osallistumista, Nuori yrittäjyys -toimintaan kuulumista, yritystä, kotisivuja, toimintaa ja yhteystietoja. Lisäksi on mahdollista poistaa nykyisiä kuvia sekä lisätä uusia kuvia. Kuvien poisto tapahtuu valitsemalla checkbox-tyyppisiä input-elementtejä, joiden nimi on del_img[]. Nimi pitää olla taulukkomuodossa, jotta pystytään käsittelemään useita kuvia lomakkeen prosessointitiedostossa. Tässä käsittelytiedostossa myös varmistetaan, että sessiomuuttujassa oleva user_id-arvo täsmää projektin host_id-kentän arvon kanssa.

Kuviossa 36 on projektinmuokkausnäkyminen projektin nimen ja lyhyen kuvauksen osalta. Sivustolla on käytössä tiettyihin textarea-elementteihin merkkilaskuri, joka toteutetaan JavaScript-puolella length-ominaisuutta hyödyntäen. Kuviossa 37 on kuvien poistonäkyminen. Muutokset tallennetaan eli lomake lähetetään painamalla ”Tallenna muutokset” -nappia, joka on submit-tyyppinen input-elementti.

KUVIO 36. Projektin muokkaus - projektin nimi, projektin lyhyt kuvaus



Poista kuva

KUVIO 37. Projektin muokkaus - kuvien poisto

4.5.3 Sosiaalinen media

Projekteja on mahdollista jakaa eri sosiaalisen median palveluihin. Edellisessä versiossa olleen Facebookin tykkää-napin lisäksi käyttäjällä on nyt mahdollisuus jakaa projekti twitteriin, StumbleUponiin, Deliciousiin ja Digiin. Twitteriin jakamiseen on kaksi tapaa: twitterin omaan rajapintaan perustuva laskurin sisältävä nappi sekä itse tehty nappi. Facebookiin jakaminen onnistuu lisäksi Facebookin käyttämän sharer.php:n avulla, joka ottaa parametreinaan osoitteen (u) ja otsikon (t).

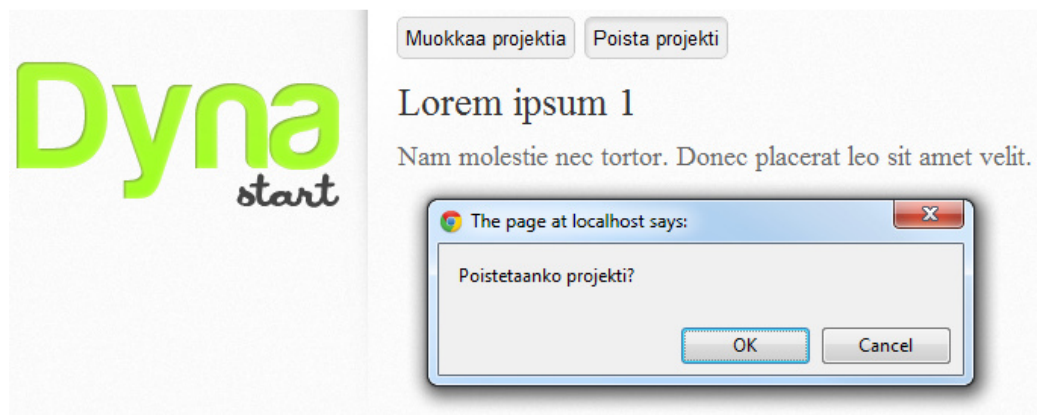
Kuviossa 38 on sosiaalisen median jakamisnapit sivustolla. Ne löytyvät sekä projektin pitkän kuvauksen yläpuolelta sekä pitkän kuvauksen lopusta tai kuvien lopusta. Koodipuolella jakamisosoitteessa olevien parametrien arvojen formaatti pitää olla URL-enkoodattu, joka saadaan aikaseksi PHP:n urlencode-funktiolla. Se on sama muoto, jolla lomakkeesta lähetetty data enkoodataan.



KUVIO 38. Projektin jakaminen sosiaalisen median palveluihin

4.5.4 Poisto

Omia projekteja on mahdollista poistaa. Poisto on toteutettu samalla periaattella kuin käyttäjätlinkin poisto. ”Poista projekti” -nappia painettaessa kysytään JavaScriptin confirm-metodilla, haluaako käyttäjä poistaa projektin. OK-nappia painettuaan poistetaan kaikki kyseisen projektin viitteet tietokannasta sekä mahdolliset fyysiset kuvat. Tämän jälkeen tulee ilmoitus projektin poistamisesta, ja käyttäjä ohjataan sivuston etusivulle. Kuviossa 39 on tilanne, jossa käyttäjä on painanut projektin poistonappia.



KUVIO 39. Projektin poistovarmistus

4.6 Kielivalinta

Sivustoa on mahdollista käyttää sekä suomen- että englanninkielellä. Oletuksena käytettävä kieli on suomi. Käyttäjä voi vaihtaa kielen sivustolla olevista kovalinkeistä.

Merkkijonot tulostetaan taulukkomuuttujasta identifier-kentän perusteella, johon haetaan tietokannasta haluttu kielikenttä. Englannin kielen ollessa valittuna sivuston osoiterivissä kulkee mukana GET-parametri ”lang”, jonka arvo on ”en”. Jos taas sivustoa käytetään suomeksi, ei tätä parametria käytetä ollenkaan. Static_strings-taulun kenttä value sisältää sivuston suomenkieliset osuudet, ja value_en sisältää puolestaan sivuston englanninkieliset osuudet.

Samaa periaattetta käytetään myös usein kysytyjen kysymysten (FAQ) -sivulla sekä kategorioita tulostettaessa. Kuviossa 40 on sekä kielimerkkijonojen että kategoriamerkkijonojen tietokantakyselyt sekä haku taulukkomuuttujaan.

Kielimerkkijonojen osalta avaimena toimii muuttujan ensimmäinen alkio (0) eli identifier-kenttä, jonka perusteella arvo (alkio 1) tulostetaan (value tai value_en).

Alkio 1:n arvo riippuu siitä, onko GET-parametri ”lang” asetettu arvolla ”en” (sivusto englanniksi) vai käytetäänkö sitä suomeksi (”lang” ei käytössä).

```

16 // kategorioiden enkkustringit, static strings en
17 if(isset($_GET['lang']) && $_GET['lang'] == "en")
18 {
19     $cat_result = mysql_query("SELECT category_id, category_en FROM categories");
20     $lang_result = mysql_query("SELECT identifier, value_en FROM static_strings");
21 }
22 // kategorioiden suomistringit, static strings fin
23 else
24 {
25     $cat_result = mysql_query("SELECT category_id, category FROM categories");
26     $lang_result = mysql_query("SELECT identifier, value FROM static_strings");
27 }
28 $i = 1;
29 while($cat_row = mysql_fetch_array($cat_result))
30 {
31     $inc_categories[$i] = array($cat_row[0], $cat_row[1]);
32     $i++;
33 }
34 $y = 0;
35 while($lang_row = mysql_fetch_array($lang_result))
36 {
37     $ss[$y] = array($lang_row[0] => $lang_row[1]);
38     $y++;
39 }

```

KUVIO 40. Sivuston merkkijonojen toiminnan periaate

5 YHTEENVETO

Alkuperäistä suunnitelmaa tarkastellessa sivuston keskeisimmät toiminnot ja ulkonäkö saatiin enemmän siihen muotoon, kun ne asiakas oli alunperin halunnut. Projektin työnjako ohjelmoijan ja visualistin välillä toimi odotetusti, ja saatiin muodostettua kokonaisuus, joka käytettävän ajan puitteissa onnistuneesti toteutettiin. Ohjelmoija joutui hieman ottamaan kantaa myös CSS-tyylimäärityksiin, mutta pääasiallisesta ilmeestä vastasi visualisti.

Aikataulua tarkasteltaessa alkukesä meni uuden sivuston suunnitteluun (asiakasvaatimukset) ja edellisen toteutuksen tutustumiseen. Kesällä tehtiin suurin osuus käytännön osalta, ja loppukesä sekä syksy keskittyi enimmäkseen viimeistelyyn. Ideointivaiheessa esiin tulleet ideat priorisoitiin onnistuneesti ja kaikki keskeisimmät ominaisuudet saatiin koottua yhteen muottiin.

Käytetyt tekniikat soveltuivat hyvin toteutuksen tekemiseen, ja varsinkin jQuery selvensi ja yksinkertaisti asiakaspuolen toimintojen toteuttamista. Osa toteutettavista toiminnoista vaati syvempää perehtymistä, ja aikaa kului myös esimerkiksi merkkijonojen syöttämiseen tietokantaan. Keskeisimmät ongelmat saatiin ratkaistua, ja tämän myötä myös henkilökohtainen tietotaito kohentui. Jälkeenpäin mietittynä joitain tehtyjä ratkaisuja olisi voinut yksinkertaistaa tai yhdistää sekä mahdollisesti rakentaa sivusto esimerkiksi jonkin kehyksen ympärille (CakePHP, yms.) toimintomäärän kasvaessa. Myös tietokantataulujen rakenteen suunnittelu olisi vaatinut perusteellisempaa tarkastelua.

Sivusto on nyt julkisesti toiminnassa osoitteessa dynastart.fi. Ensimmäiset käyttökokemukset ja palautteet ovat olleet positiivisia, ja kiinnostus tämän tyyppiseen konseptiin on ilmeinen. Tulevaisuudessa sivustoa tullaan yhä kehittämään nykyisten toimintojen ja uusien toimintojen muodossa. Uusia ajankohtaisia kohteita ovat muun muassa keskustelufoorumi, yksityisviestit sekä Facebook-rekisteröinti. Käyttäjäkunnan ja toimintojen kasvaessa myös palvelimen pitää vastata kapasiteetiltaan sen hetkiseen tarpeeseen.

LÄHTEET

Bibeault, B. & Katz, Y. 2008. jQuery in Action. Greenwich: Manning Publications Co.

Chaffer, J. & Swedberg, K. 2007. Learning jQuery. Birmingham: Packt Publishing Ltd.

Darie, C., Brinzarea, B., Chereches-Tosa, F. & Bucica, M. 2006. AJAX and PHP. Birmingham: Packt Publishing Ltd.

Oracle. 2011. Using the Event Scheduler [viitattu 17.10.2011]. Saatavissa: <http://dev.mysql.com/doc/refman/5.1/en/events.html>

Rose India. 2008. What is jQuery? [viitattu 13.10.2011]. Saatavissa: <http://www.roseindia.net/ajax/jquery/whatisjquery.shtml>

The jQuery Project. 2010. Documentation [viitattu 14.10.2011]. Saatavissa: <http://docs.jquery.com>

Tutorials Point. 2011. HTTP Protocol [viitattu 11.10.2011]. Saatavissa: <http://www.tutorialspoint.com/http/index.htm>

W3Schools. 2011a. AJAX Tutorial [viitattu 13.10.2011]. Saatavissa: <http://www.w3schools.com/ajax/default.asp>

W3Schools. 2011b. HTML DOM Tutorial [viitattu 12.10.2011]. Saatavissa: <http://www.w3schools.com/html/dom/default.asp>

W3Schools. 2011c. jQuery Tutorial [viitattu 13.10.2011]. Saatavissa: <http://www.w3schools.com/jquery/default.asp>

Wellman, D. 2011. jQuery UI 1.8. Birmingham: Packt Publishing Ltd.

Wikipedia. 2011. CGI [viitattu 11.10.2011]. Saatavissa:

<http://fi.wikipedia.org/wiki/CGI>

Zaefferer, J. 2009. jQuery plugin: Validation [viitattu 14.10.2011]. Saatavissa:

<http://bassistance.de/jquery-plugins/jquery-plugin-validation/>