

WWW-sovellukset, harjoitustyö

# Jalkapallokenttien varausjärjestelmä

Dokumentaatio

Konsta Ala-Ilomäki 0418514

12-12-2016

## Sisällysluettelo

1 Yleistä.....	2
2 Rakenne .....	3
3 Ominaisuudet ja pisteytys .....	4
3.1 Responsiivinen ulkoasu (5p).....	4
3.2 Tietokannan käyttö tietojen tallennukseen (5p) .....	4
3.3 Käyttäjän autentikointi (5p) .....	4
3.4 Front-controllerin käyttö (3p) .....	4
3.5 MVC-mallin mukainen sivusto (5p) .....	4
3.6 JSONin käyttö tiedon liikuttelussa (3p).....	5
3.7 jQueryn käyttö (3p).....	5
3.8 AJAX-ohjelmoinnin hyödyntäminen (3p).....	5
3.9 Kattava dokumentaatio (5p) .....	5
3.10 Kolmannen osapuolen palvelun käyttö (5p).....	5
4 Ympäristö- ja kirjastovaatimukset .....	6

# 1 Yleistä

Sovelluksen tarkoituksena on mahdollistaa eri urheiluseuroille tai muille organisaatioille jalkapallokenttien varaaminen. Sovelluksen pääominaisuuksia ovat järjestelmään syötettyjen jalkapallokenttien varaaminen korkeintaan viikon päähän, yleisen varaustilanteen tarkastelu, omien varausten tarkastelu ja kenttien tarkastelu. Lisäksi sovellukseen käyttö vaatii rekisteröitymisen ja kirjautumisen.

Mikäli käyttäjä ei ole kirjautunut sisään, hän voi ainoastaan tarkastella kenttiä. Satunnainen käyttäjä ei siis voi varata kenttiä. Rekisteröitymällä ja kirjautumalla käyttäjä voi varata kenttiä ja tarkastella omia sekä muiden varauksia. Sovelluksen eri ominaisuudet löytyvät ylävalikosta.

Aihe on valittu osittain esimerkki numero neljästä inspiroituneena. Aihe valikoitui vahingossa, kun ideat olivat vähissä. Sovelluksen toiminnallisuus on toteutettu PHP:lla ja JavaScriptillä (myös jQuery ja AJAX/AJAJ). Sovellus käyttää tietokantaa kaiken tiedon tallentamiseen (käyttäjät, varaukset, kentät). Sovelluksen luuranko on tehty HTML:llä ja tyyli käyttäen CSS-muotoilua. Sovellus hyödyntää Google Mapsin JavaScript API:a. Data siirtyy palvelimen ja clientin välillä JSON-muotoisena.

## 2 Rakenne

Sovelluksen rakentuu kolmesta tai viidestä valikon linkistä riippuen siitä onko käyttäjä kirjautuneena. Etusivulla voidaan esittää esimerkiksi ohjeita sovelluksen käyttöön tai uutisia. Kentän varaaminen -välilehti näyttää pudotusvalikossa kaikki järjestelmässä olevat kentät, joista käyttäjä voi valita yhden kerrallaan. Valittuaan kentän käyttäjä voi painaa Näytä-painiketta, joka tulostaa sivulle kalenterin ja kyseisen kentän varaustilanteen. Kysymysmerkki-painike avaa dialogin, josta käy ilmi kalenterissa esiintyvien symbolien tarkoitus. Kalenterissa esiintyvät symbolit ovat suurennuslasi, oikein-merkki, lukko sekä rasti. Suurennuslasia painamalla sovellus näyttää tietoa varauksesta. Oikein-merkki varaa kentän. Lukko tarkoittaa, että joku muu on varannut kentän. Rasti tarkoittaa, että kyseessä on käyttäjän oma varaus ja rastia painamalla varaus perutaan.

Sovellus antaa käyttäjälle palautetta jQueryn dialog-ruutujen avulla. Esimerkiksi kentän varaaminen, varauksen peruuttaminen tai varauksen tarkastelu avaavat dialog-ruudun.

Omat varaukset -välilehdellä sovellus tulostaa käyttäjälle taulukon kaikista hänen varaamistaan vuoroista. Taulukosta käy ilmi varattu kenttä, aika sekä päivämäärä. Lisäksi taulukosta löytyy painike varauksen peruuttamiseksi.

Kentät-välilehti kokoaa yhteen taulukkoon kaikki järjestelmässä olevat kentät. Taulukosta näkee sekä kentän nimen että kentän osoitteen. Taulukon rivien lopussa on karttapainike, joka avaa kartan kyseessä olevasta kentästä.

Kirjautuminen-välilehti tarjoaa vaihtoehdot sisään- ja uloskirjautumiseen sekä rekisteröitymiseen. Lisäksi kirjautunut käyttäjä näytetään valikon yläpuolella tekstinä.

### 3 Ominaisuudet ja pisteytys

Alla mainittujen ominaisuuksien lisäksi tavoitteena on viisi pistettä hyvästä vertaisarvioinnista, jolloin yhteispistemääräksi muodostuu  $5+5+5+3+5+3+3+3+5+5+5 = 47p$  eli 40p.

#### 3.1 Responsiivinen ulkoasu (5p)

Responsiivisuus näkyy, kun ikkunan leveys pienennetään 800 pikseliin tai alle. Tärkeimmät muutokset ovat animoidun jalkapallon häviäminen, logon, ”kirjautuneena”-tekstin ja valikon sulautuminen yhteen sekä valikon tiivistyminen. Lisäksi sivulla olevien elementtien maksimileveyttä on rajoitettu, jolloin ne skaalautuvat paremmin näyttöä pienentäessä.

#### 3.2 Tietokannan käyttö tietojen tallennukseen (5p)

Sovellus käyttää tietokantaa kaiken säilytettävän tiedon tallentamiseen. Tietokannasta löytyvät taulut varauksille, käyttäjille ja kentille. Tietokantaan tehdään kyselyitä PHP:n PDO:n avulla.

#### 3.3 Käyttäjän autentikointi (5p)

Käyttäjän autentikointiin käytetään PHP:n tarjoamia `password_hash()` ja `password_verify()` -funktioita. `password_hash()` lisää salasanaan suolan ja käyttää turvallisena pidettyä salausalgoritmia. Tietokantaan tallennetaan salasanat hasheina.

#### 3.4 Front-controllerin käyttö (3p)

Sovelluksessa on `index.php`-tiedosto, joka ohjaa sovelluksen suoritusta perustuen PHP:n superglobaaliin `$_GET["p"]`.

#### 3.5 MVC-mallin mukainen sivusto (5p)

Sovellus sisältää MVC-mallin mukaisesti `.php`-tiedostot eri toiminnoille. Esimerkiksi `utils.php` tulostaa käyttäjälle sivuston rungon. `Reservation.php` parsii käyttäjän syötteen ja lähettää sen edelleen esimerkiksi `makeReservation.php`:lle, joka tekee tietokantaoperaation.

### 3.6 JSONin käyttö tiedon liikuttelussa (3p)

Tietokantaoperaatioiden jälkeen PHP:n echoama tieto on aina JSON-muotoista. Käytössä on PHP:n `json_encode()` ja JavaScriptin `JSON.parse()`.

### 3.7 jQueryn käyttö (3p)

jQuery on vahvasti mukana sovelluksessa. jQueryä käytetään esimerkiksi html-elementtien muokkaamiseen ajon aikana tai dialog-ruutujen avaamiseen. Lisäksi monien painikkeiden toiminnallisuus on toteutettu jQueryllä.

### 3.8 AJAX-ohjelmoinnin hyödyntäminen (3p)

AJAX:a käytetään lähes jokaisen painikkeen kanssa, kun painike aiheuttaa tietokantaoperaation.

### 3.9 Kattava dokumentaatio (5p)

Tässä dokumentissa on mielestäni mainittu kaikki olennainen varsin tarkoin yksityiskohdin.

### 3.10 Kolmannen osapuolen palvelun käyttö (5p)

Sovelluksessa käytetään Google Mapsin JavaScript API:a, jolla ensin geokoodataan (muutetaan osoite koordinaateiksi) jalkapallokentän osoite ja sitten luodaan sivustolle kartta kohteesta

## 4 Ympäristö- ja kirjastovaatimukset

Sovellus on testattu toimivaksi Cloud 9:ssä. Cloud 9:ssä pitäisi olla valmiiksi asennettuna mySQL-tietokannat. Muita erityisiä kirjastoja ei tarvitse asentaa.

Repository <https://github.com/konala/WWW-sovellukset>

Ohjeet sovelluksen käyttöön:

### Tapa 1

1. Luo Cloud 9:ssä uusi workspace ja valitse templateksi **"PHP, Apache & MySQL"**.
2. Aja bashissa komento  
**git clone https://github.com/konala/WWW-sovellukset.git**
3. Aja bashissa seuraavat komennot järjestyksessä:
  - a. **mysql-ctl start**
  - b. **mysql-ctl cli**
  - c. **use c9;**
  - d. **source WWW-sovellukset/c9/database.sql;**
4. Paina **"Run Project"**
5. Avaa preview tai linkki sovellukseen (muotoa <https://workspacen-nimi-käyttäjänimi.c9users.io>) ja navigoi **WWW-sovellukset/c9/**
6. Sovellus on käyttövalmis

### Tapa 2

Käytä julkista workspacea, jossa kaikki on valmiina. Tarvittaessa sinun täytyy pyytää käyttö lupaa erikseen. Linkki workspaceen: <https://ide.c9.io/konstaai/www-sovellukset-ht#openfile-README.md>

Tietokannassa on valmiina yksi testikäyttäjä. Testikäyttäjän sähköposti on [erkki@esimerkki.fi](mailto:erkki@esimerkki.fi) ja salasana on "salasana".