TIKO 2017 HARJOITUSTYÖ

2. vaihe

Mark Mäkinen & Jukka Lehtimäki, ryhmä 38

Sisältö

Järjestelmä

“SQL-treeni” -järjestelmä on helposti käytettävissä osoitteessa https://zini.im/tiko. Vaikkakin helppoussyistä järjestelmää ei jaella yliopiston palvelimelta, sen toteutuksessa ei ole käytetty mitään mitä ei löytyisi yliopiston palvelimelta. Tietokantana toimii PostgreSQL, kuten vaadittu.

Järjestelmän lähdekoodi on saatavilla GitHub-palvelussa osoitteessa https://github.com/yakka34/tiko2017. Käyttäjävastaavuudet: Z1ni – Mark Mäkinen, yakka34 – Jukka Lehtimäki.

Järjestelmässä on oletuksena tehtävänannossa annettu aineisto, sekä neljä käyttäjää:

- Pääkäyttäjä: admin@example.com
- Opettaja: opettaja1@example.com
- Oppilas 1 *: luigi24@example.org
- Oppilas 2: * gislason.darren@example.net

Kaikkien käyttäjien salasana on ”salasana”.

***)** Nämä sähköpostiosoitteet ovat voimassa osoitteessa https://zini.im/tiko sijaitsevalla testipalvelimella kirjoitushetkellä. Osoitteet ovat saattaneet voida muuttua kirjoitushetken jälkeen korjausten vuoksi. Nämä osoitteet eivät myöskään ole voimassa, mikäli luodaan oma lokaali asennus. Käyttäjien osoitteet voidaan tarkistaa pääkäyttäjänä hallintasivulta (kirjaudu -> oikea yläkulma -> Hallinta), koska pääkäyttäjän ja opettajan sähköpostit ovat aina yllä mainitut.

Ominaisuudet

Lyhyt ominaisuuslistaus ominaisuudet-liitteessä.

Sovellus mahdollistaa tehtävänannossa vaaditulla tavalla tehtävien ja tehtävälistojen luomisen, jotka sisältävät SQL-kyselyharjoituksia. Opiskelijat voivat kirjautua sisään ja suorittaa tehtäviä ja saada niistä palautetta. Opettajat voivat luoda tehtäviä ja tehtävälistoja (tehtäväkokonaisuuksia) ja tarkastella suoritusraportteja ja niiden oppilaiden tietoja, jotka ovat tehneet opettajan luomia tehtäviä. Pääkäyttäjällä on oikeus tarkastella kaikkien käyttäjien tietoja. Järjestelmä toimii sekä rooli- että oikeustasolla. Käyttäjillä on rooli ja rooleihin kuuluu useampia oikeuksia (esim. oikeus luoda tehtäviä, oikeus tarkastella käyttäjien tietoja, jne.).

Rakenteellisesti järjestelmässä on tehtäväsuorituksia, jotka liittyvät sessiosuorituksiin, jotka liittyvät sessioihin. Tämä toteuttaa tehtävänannon T1- ja T2-tapahtumat/tapahtumasekvenssit.

Tehtävänannossa mainituista raporteista on toteutettu kaikki muut paitsi raportti R6. Raportit löytyvät pääkäyttäjän ja opettajan etusivulta.

Tietokannan rakenteen johtamista ja esittämistä ei ole toteutettu, vaan sen katsotaan toimivan samalla tavalla esimerkkijärjestelmän lailla (Tiko-kurssin SQL-kyselyjärjestelmä), eli opettajan pitää sisällyttää kannan rakenne ja tiedot tehtävälistan/tehtävän kuvaukseen. Kuvaustekstejä editoidaan monipuolisella editorilla, joka mahdollistaa kehittyneet tekstinkäsittelyominaisuudet sekä kuvien liittämisen kuvauksiin.

Järjestelmä tukee useita samanaikaisia käyttäjiä ja kestää virheellisesti suoritetut, tietokannan tilaa muuttavien toimintojen suorittamisen. Käyttäjä voi esimerkiksi virheellisesti poistaa kaikki tiedot esimerkkitaulusta, jolloin järjestelmä palauttaa kaikkien taulujen tiedot kyselyä edeltävään tilanteeseen.

Käyttöliittymä on toteutettu visuaalisesti miellyttäväksi käyttäen Bootstrap-kehystä. Lisäksi tehtävälistojen luominen on toteutettu JavaScriptillä dynaamisesti (ks. oikea yläkulma -> Tehtävähallinta).

Pääkäyttäjällä on oikeus ja mahdollisuus muuttaa käyttäjien tietoja ja lisätä käyttäjille rooleja.

Toteutus

Järjestelmä on toteutettu Laravel-ohjelmistokehyksellä MVC-mallia käyttäen. Laravelin käyttämisestä johtuen tietokannan malli on jokseenkin erilainen kuin 1. vaiheen suunnitelmassa. Ydinrakenne ja toiminnallisuus on kuitenkin käytännössä sama. Tietokannan rakenteen voi korkeammalla tasolla tarkistaa lähdekoodista ns. migration-tiedostoista, jotka sijaitsevat kansiossa src/database/migrations.

Järjestelmän ydinrakenteena ovat tehtävät, tehtävälistat, sessiot ja tehtäväyritykset. Tehtävät sisältävät SQL-kyselyitä ja niihin tehtävänantoja. Tehtävät kuuluvat tehtävälistoihin, jotka sisältävät yhden tai useamman tehtävän ja kuvauksen. Käyttäjät voivat tehdä tehtäviä, jotka tällöin käsitellään sessioina. Sessiossa on aina yksi tehtävälista, jonka tehtävät käyttäjän pitää suorittaa joko oikein kerralla tai väärin kolme kertaa per tehtävä. Sessiossa yhdellä tehtävällä on siis yksi tai maksimissaan kolme tehtäväyritystä. Sessioista ja tehtäväyrityksistä pidetään myös ajallisesti kirjaa, kuten tehtävänannon raporteista käy ilmi.

Työnjako

Työ on toteutettu hyvin tasaisesti kaksin, eikä kumpikaan ole tehnyt vain mitään tiettyjä osia. Tarkemmat graafit ja commit-määrät voi tarkistaa GitHub-palvelusta osoitteesta https://github.com/yakka34/tiko2017/graphs/contributors.

Käyttö

Asennus

Jos järjestelmän haluaa asentaa palvelinkoneelle, vaaditaan koneelta seuraavaa (tässä vaiheessa kannattaa muistaa, että täysin toimiva ja käytettävä versio on käytettävissä osoitteessa https://zini.im/tiko):

- PHP 7
- PHP Composer
- Node.js 6.x

Asennus suoritetaan kloonaamalla git-repo, luomalla .env-asetustiedosto ja ajamalla src-kansiossa muutama komento. src/public -kansio pitäisi linkata symbolisella linkillä kansioon, josta verkkopalvelin voi tarjota sivuja (ks. ln-komento linuxilla).

.env-tiedosto

Kopioi .env.example-tiedosto .env-tiedostoksi ja muuta seuraavat kohdat: APP_URL, DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, DB_PASSWORD (myös DB_TASK_*-vastaavat). Mitään asetuksia ei tarvitse muuttaa, jos sovellusta ajetaan Homestead-virtuaalikoneessa (https://laravel.com/docs/5.4/homestead)

Komennot (konsolissa)

```
$ composer install
$ php artisan key:generate
$ php artisan migrate –seed
$ npm install
$ npm run prod
```

Käyttö

Järjestelmä on websovellus, jota käytetään selaimessa. Mikäli on asennettu oma lokaali versio järjestelmästä Homestead-virtuaalikoneeseen, sovellukseen pääsee kiinni Homesteadin konfiguraatiossa määritellyllä osoitteella. Vapaasti verkossa saatavilla oleva testiversio on osoitteessa https://zini.im/tiko.

Järjestelmään kirjaudutaan oikeasta yläkulmasta linkistä ”Kirjaudu”. Tunnukset näkyvissä ylempänä kohdassa Järjestelmä. Kirjautumisen jälkeen toimintoja löytyy etusivulta ja oikeasta yläkulmasta käyttäjänimeä klikkaamalla. Käyttöliittymä on yksinkertainen ja toivottavasti selkeästi navigoitavissa, joten tässä ei kuvata toimintoja tämän enempää. Järjestelmään toteutetut toiminnot voi tarkastaa ylempää kohdasta Ominaisuudet.

Tehtävälistoja pääsee suorittamaan opiskelijana kirjautumisen jälkeen etusivulta klikkaamalla tehtävälistan nimeä.

Arvio työstä

Järjestelmän rakentaminen on ollut sopivan haastavaa, varsinkin kun samaa järjestelmää on käytetty myös TIETA12 WWW-ohjelmointi -kurssin harjoitustyönä. Mikään kohta ei ole erityisesti noussut vaikeana esille, vaan kokonaisuutena järjestelmän rakentaminen on ollut haastavaa. Eri osien toimiminen yhteen on tästä hyvä esimerkki, varsinkin kun järjestelmää on tehty parityönä eri osia tehden. Kommunikointi on ollut hyvin tärkeää sekä tekstuaalisesti, että audion välityksellä.

Puutteina mainittakoon esimerkiksi järjestelmän suhteellinen joustamattomuus. Tavoitteena oli tehdä tietokantahiekkalaatikosta (jossa siis tarkastetaan käyttäjän syöttämät SQL-lauseet) paljon joustavampi järjestelmä, mutta ajankäyttö ja priorisointi aiheuttivat sen, että harjoitustyön esimerkkikantojen tila kovakoodattiin hiekkalaatikkojärjestelmään. Tämä ei kuitenkaan ole ongelma tämän työn kontekstissa. Toisena puutteena sanottakoon tietojen poistamisen mahdottomuus. Mitään tehtäviä tai käyttäjiä, jne. ei voi poistaa järjestelmästä. Tämäkään ei kuitenkaan ole tällä hetkellä ongelma tämän harjoitustyön puitteessa.
Järjestelmä ei myöskään estä kyselyitä muihin kuin tehtävänannossa mainittuihin tauluihin. Tässä kohtaa toivotaan, että järjestelmän ylläpitäjä käyttäisi hiekkalaatikkotietokantana erillistä tietokantaa (tämä on tällä hetkellä mahdollista konfiguraation puolesta) ja loisi oman käyttäjän, jolla olisi vain SELECT, UPDATE, INSERT ja DELETE -oikeudet.

Mainitut ongelmat ovat siis hypoteettisen jatkokehityksen, mutta eivät harjoitustyön kannalta oleellisia.

Liite – Ominaisuuslomake

Tietokantaohjelmointi 2017						
Lomake toteutetuista harjoitustyön ominaisuuksista						
Rastita ominaisuudet, joita työssä on. Arvioi onko *) merkityt ominaisuudet toteutettu suppeasti vai laajasti.						
Ryhmän numero:						
				Toteutettu		
	Toteutettu			laajasti		
Tapahtuma T1	**X**					
Tapahtuma T2	**X**					
Raportti R1	**X**					
Raportti R2	**X**					
Raportti R3	**X**					
Raportti R4	**X**					
Raportti R5	**X**					
Raportti R6						
Tietokannan rakenteen johtaminen ja esittäminen käyttäjille						
Tuki samanaikaiselle käytölle *)	**X**			**X**		
Käyttöliittymäominaisuudet *)	**X**			**X**		
Käyttäjäryhmät *)	**X**			**X**		
Mahdollisia lisätoimitoja (mitä?) *)						
Lisätietoa						