

Laravel::DB::ScoreTronic01 - autentikoinnin perusteet

Tässä ja tulevissa luvuissa esitetään ja toteutetaan osa opintojaksolla käytössä olevan ScoreTronic-sovelluksen toiminnallisuudesta. Tutustumisen yhteydessä on tärkeää, että itse käytännössä kokeilet ja mahdollisesti muuntelet annettuja esimerkkejä hallitaksesi sisällön. Tarkoituksena on havainnollistaa seuraavia asioita

- tietokannan luomista/käyttämistä ilman migraatioita ja Laravelin sidonnaisuuksia (taulun nimeämiset, moni-moneen-yhtedet jne) siten että voisit hyödyntää esim. aiemmin MySQL WorkBenchillä luomaasi tietokantaa ja aiemmin hankittua SQL-osaamista sellaisenaan.
- Laravelin autentikointi- ja autorisointimekanismeja
- CSV-tiedoston datan lataamista tietokantaan
- Sessiomuuttujien käyttöä virheilmoitusten yhteydessä
- Transaktiomekanismin ja poikkeusten käsittelyn hyödyntämistä.

Esitietoina ja -valmisteluina olet tehnyt aiemmin esitetyn StudeTronic-sovelluksen tietokanta-asetuksineen. Erityisesti käyttäjät tallennetaan tässä sovelluksessa Laravelin oletusarvoiseen `users`-tauluun, joka tulisi olla jo valmiina Taulussa ei ole kuitenkaan dataa vielä tallennettuna:

```
mysql> show tables;
+-----+
| Tables_in_laraveldb |
+-----+
| course               |
| customers             |
| failed_jobs          |
| migrations           |
| password_resets      |
| student              |
| student_has_course   |
| users                |
+-----+
8 rows in set (0.00 sec)
```

```
mysql> desc users;
+-----+-----+-----+-----+-----+
| Field          | Type                | Null | Key | Default |
+-----+-----+-----+-----+-----+
| id             | bigint(20) unsigned | NO   | PRI | NULL    |
| name           | varchar(255)        | NO   |     | NULL    |
| email          | varchar(255)        | NO   | UNI | NULL    |
| email_verified_at | timestamp          | YES  |     | NULL    |
| password       | varchar(255)        | NO   |     | NULL    |
| remember_token | varchar(100)        | YES  |     | NULL    |
| created_at     | timestamp          | YES  |     | NULL    |
| updated_at     | timestamp          | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
```

```

+-----+-----+-----+-----+
8 rows in set (0.00 sec)

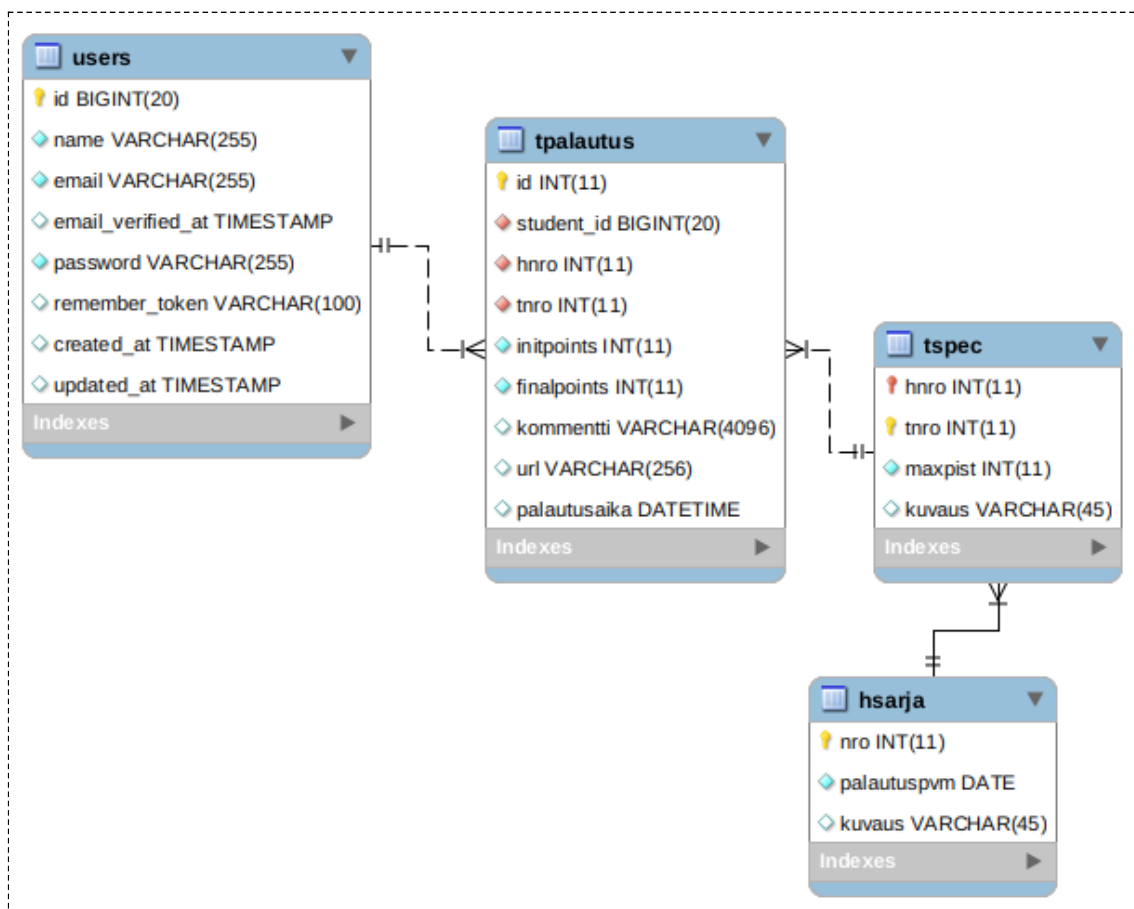
mysql> select * from users;
Empty set (0.00 sec)

mysql>

```

Tietokannan rakenne

Luotavan ja muokattavan tietokannan lopullinen rakenne tulee olemaan seuraava



Huomioita

- **users**-taulu syntyy Laravelin migraatiotiedoston perusteella. Tieto rekisteröityneistä käyttäjistä (tehtäviä palauttavista oppilaista + admin-käyttäjät) tallennetaan **users**-tauluun.
- **hsarja**-taulussa on tallennettuna yhden harjoitustehtäväsarjan (Esim Harjoitukset 2) tiedot palautuspäivämäärineen.
- **tspec**-taulussa on tallennettuna yhden tiettyyn harjoitussarjaan kuuluvan tehtävän (Esim Harjoitukset 2 Tehtävä 1) tiedot. Taulussa on tallennettuna tieto kunkin tehtävän maksimipisteistä
- **tpalautus**-taulussa on tallennettuna oppilaiden tehtäväpalautukset ja opettajan hyväksymät pisteet
- **tpalautus**-taulun **student_id**-viiteavain **users**-taulun **id**-pääavaimeen luodaan viimeisenä toimenpiteenä **ALTER TABLE** -lauseella

36.01 Luodaan tietokannan taulut

`users`-taulua lukuunottamatta sovelluksen tietokannan taulut luodaan käyttämättä Laravelin migraatiotiedostoja. Ohessa on esitetty näin luotavien taulujen `CREATE TABLE` -lauseet. Voit valintasi mukaan luoda taulut **laraveldb**-tietokantaasi **mysql**-komentokehityksessä `mysql >`, **MySQL WorkBench** -ohjelmalla tai muulla haluamallasi tavalla.

```
1  -- -----
2  -- Table `laraveldb`.`hsarja`
3  -- -----
4  CREATE TABLE IF NOT EXISTS `laraveldb`.`hsarja` (
5      `nro` INT NOT NULL,
6      `palautuspvm` DATE NOT NULL,
7      `kuvaus` VARCHAR(45) NULL,
8      PRIMARY KEY (`nro`))
9  ENGINE = InnoDB;
10
11
12  -- -----
13  -- Table `laraveldb`.`tspec`
14  -- -----
15  CREATE TABLE IF NOT EXISTS `laraveldb`.`tspec` (
16      `hnro` INT NOT NULL,
17      `tnro` INT NOT NULL,
18      `maxpist` INT NOT NULL,
19      `kuvaus` VARCHAR(45) NULL,
20      PRIMARY KEY (`hnro`, `tnro`),
21      CONSTRAINT `fk_tspec_hsarja1`
22          FOREIGN KEY (`hnro`)
23          REFERENCES `laraveldb`.`hsarja` (`nro`)
24          ON DELETE NO ACTION
25          ON UPDATE NO ACTION)
26  ENGINE = InnoDB;
27
28
29  -- -----
30  -- Table `laraveldb`.`tpalautus`
31  -- -----
32  CREATE TABLE IF NOT EXISTS `laraveldb`.`tpalautus` (
33      `id` INT NOT NULL AUTO_INCREMENT,
34      `student_id` BIGINT UNSIGNED NOT NULL,
35      `hnro` INT NOT NULL,
36      `tnro` INT NOT NULL,
37      `initpoints` INT NOT NULL,
```

```

38 `finalpoints` INT NOT NULL,
39 `kommentti` VARCHAR(4096) NULL,
40 `url` VARCHAR(256) NULL,
41 `palautusaika` DATETIME NULL,
42 PRIMARY KEY (`id`),
43 INDEX `fk_tpalautus_tspec1_idx` (`hnro` ASC, `tnro` ASC),
44 CONSTRAINT `fk_tpalautus_tspec1`
45 FOREIGN KEY (`hnro` , `tnro`)
46 REFERENCES `laraveldb`.`tspec` (`hnro` , `tnro`)
47 ON DELETE NO ACTION
48 ON UPDATE NO ACTION)
49 ENGINE = InnoDB;

```

Viiteavain tpalautus-tauluun

Luodaan lopuksi tpalautus-taulun student_id-viiteavain users-taulun id-pääavaimeen ALTER TABLE -lauseella

```

1 ALTER TABLE tpalautus
2 ADD CONSTRAINT `fk_tpalautus_users`
3 FOREIGN KEY (`student_id`)
4 REFERENCES `users` (`id`)
5 ON DELETE RESTRICT
6 ON UPDATE RESTRICT;

```

Luodaan autentikointiin tarvittavat tiedostot

Laravelin rekisteröinnin ja kirjautumisen käyttöönotossa luodaan näkymät eli template-tiedostot, jotka käyttävät luomaansa resources/views/layouts/app.blade.php-tiedostoa. Jos haluat säilyttää järjestelmässäsi aiemman app.blade.php-tiedoston, (varmuus)kopioi se vaikkapa nimelle resources/views/layouts/app-original.blade.php eli

```
resources/views/layouts/app.blade.php -> resources/views/layouts/app-original.blade.php
```

36.02 Luodaan tiedostot rekisteröinti- ja kirjautumistoimintoja varten

```

cd ~/public_html/projekti01
composer require laravel/ui --dev
php artisan ui bootstrap --auth

```

Jos et poistanut `layouts/app.blade.php`-tiedostoa, joudut nyt ylikirjoittamaan sen vastaamalla **yes** seuraavaan kysymykseen

```
The [layouts/app.blade.php] view already exists. Do you want to overwrite it?  
>
```

Lopuksi asennetaan `npm`-ohjelma, jolla luodaan ja asennetaan tarvittavat tiedostot. Asennettavaa on aika paljon. Asennuksen aikaiset virheilmoitusten ei pitäisi haitata tässä etenevää asennusprosessia

```
cd ~/public_html/projekti01  
sudo apt install npm  
npm install && npm run dev
```

Varmistatetaan tietokantamigraatioiden ajantasaisuus ja `users`-taulun luonti tietokantaan

```
php artisan migrate
```

Tutkitaan syntyneitä autentikaatitiedostoja

Tutkitaan syntyneitä autentikaatitiedostoja

36.03 Reittitiedoston merkinnät

Reittitiedostoon `routes/web.php` generoituu merkinnät:

```
1 | ...  
2 | Auth::routes();  
3 | Route::get('/home', 'HomeController@index')->name('home');
```

Huom: `Auth::routes()` generoi kaikki tarvittavat autentikaatioreitit
<https://stackoverflow.com/questions/39196968/laravel-5-3-new-authroutes>

36.04 Luotiin uusi controller

```
app/Http/Controllers/HomeController.php
```

36.05 Luotiin tiedostoja

resources/views/auth-kansioon mm.

- login.blade.php
- register.blade.php

ja

app/Http/Controllers/Auth-kansioon mm.

- LoginController.php
- RegisterController.php

ja

app/Http/Middleware-kansioon

Huomaa esim. salasanan minimipituuden vaatiminen

app/Http/Controllers/Auth/RegisterController.php

```
'password' => 'required|string|min:8|confirmed',
```

36.06 Autentikointia varten luotujen tiedostojen poistaminen

Joskus on vielä tässä vaiheessa tarve miettiä asioita uudestaan ja peruuttaa aikaan ennen autentikointitiedostojen luontia. Asiaa on käsitelty [tässä](#) keskustelussa,

Käyttöliittymän muokkaaminen

Tässä esitetyksessä korvataan/muokataan muutamia käyttöliittymään liittyviä tiedostoja halutun lopputuloksen aikaansaamiseksi. Muutokset ovat aika pieniä, mutta ohjeissa annetaan aina koko muutettu tiedosto, jotta virheiden mahdollisuus minimoituu.

35.07 Bootstrap-kirjastojen käyttöönotto näkymissä

Kopioidaan tässä seuraavat Bootstrap-tiedostot oheisiin sijainteihin Laravelin käyttöön (samat kuin StudeTronicin kanssa, mutta nyt eri nimelle)

`app-studetronic.css` --> `public/css/app.css`

`app-studetronic.js` --> `public/js/app.js`

36.08 Muokkaa mastertemplate-tiedostoa

Muokataan Bootstrap-kirjastoon perustuva masterlayout korvaamalla luotu `resources/views/layouts/app.blade.php`-tiedosto kokonaan seuraavalla sisällöllä. Halutessasi voit ottaa aiemmasta `app.blade.php`-tiedostosta varmuuskopioin jollekin toiselle nimelle.

`resources/views/layouts/app.blade.php`

```
1 <!doctype html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6
7     <!-- CSRF Token -->
8     <meta name="csrf-token" content="{{ csrf_token() }}">
9
10    <title>{{ config('app.name', 'Laravel') }}</title>
11
12    <!-- Scripts -->
13    <script src="{{ asset('js/app.js') }}" defer></script>
14
15    <!-- Fonts -->
16    <link rel="dns-prefetch" href="//fonts.gstatic.com">
17    <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">
18
19    <!-- Styles -->
20    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
21 </head>
22 <body>
23     <div id="app">
24         <nav class="navbar navbar-default navbar-static-top">
25
26             <div class="container">
27                 <div class="navbar-header">
28
29
30                     <a class="navbar-brand" href="{{ url('/') }}">
31                         ScoreTronic
```

```

32     </a>
33     <a class="navbar-brand" href="{{ url('/home') }}">
34         /home
35     </a>
36 </div>

37
38
39     <div class="collapse navbar-collapse" id="navbarSup
40
41     <!-- Left Side Of Navbar -->
42     <ul class="nav navbar-nav">
43
44         <li class="dropdown">
45             <a class="dropdown-toggle" data-toggle="dropdown
46             <ul class="dropdown-menu" aria-labelledby="download
47             <li><a href="{{ url('/users') }}">Opiskelijalista</
48             <li class="divider"></li>
49             <li><a href="{{ url('/harkat') }}">Harkkapalautukse
50             <li class="divider"></li>
51             <li><a href="{{ url('/toiminto22') }}">Toiminto 22<
52             <li class="divider"></li>
53             <li><a href="{{ url('/toiminto33') }}">Toiminto 33<
54
55         </ul>
56     </li>
57 </ul>

58
59
60     <!-- Right Side Of Navbar -->
61     <ul class="nav navbar-nav navbar-right">
62         <!-- Authentication Links -->
63         @guest
64             <li class="nav-item">
65                 <a class="nav-link" href="{{ route(
66             </li>
67             @if (Route::has('register'))
68                 <li class="nav-item">
69                     <a class="nav-link" href="{{ rd
70                 </li>
71             @endif
72             @else
73                 <li class="nav-item dropdown">
74                     <a id="navbarDropdown" class="nav-l
75                         {{ Auth::user()->name }} <span
76                     </a>
77

```



```

78         <div class="dropdown-menu dropdown-
79             <a class="dropdown-item" href="
80                 onclick="event.preventDefault
81                     document.getElementById
82                         {{ __('Logout') }}"
83             </a>
84
85             <form id="logout-form" action="
86                 @csrf
87             </form>
88         </div>
89     </li>
90     @endguest
91 </ul>
92 </div>
93 </div>
94 </nav>
95
96     <main class="py-4">
97         @yield('content')
98     </main>
99 </div>
100 </body>
101 </html>

```

36.09 Muokkaa sovelluksen etusivu

Muuta sovelluksesi etusivuksi <http://192.168.1.126/~testi/projekti01/public/> asettamalla tiedoston `resources/views/welcome.blade.php` sisällöksi

`resources/views/welcome.blade.php`

```

1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container">
5          <div class="col-md-8 col-md-offset-2">
6              <div class="panel panel-default">
7                  <div class="panel-heading">Dashboard</div>
8
9                  <div class="panel-body">
10                     @if (session('status'))
11                         <div class="alert alert-success" role="aler
12                             {{ session('status') }}
13                     </div>

```

```

14         @endif
15
16         ScoreTronic - Root folder [ / ]<br>
17         Login is not required to view this page
18
19     </div>
20 </div>
21 </div>
22 </div>
23 @endsection

```

Huom: Tarkista että reittitiedostossa on vielä tallella opintojakson alussa asetettu reitti:

```

1 ...
2 Route::get('/', function () {
3     return view('welcome');
4 });

```

36.10 Tarkista /home-sivun näkymä

Muuta reitin <http://192.168.1.126/~testi/projekti01/public/home> näkymätiedoston `resources/views/home.blade.php` sisällöksi seuraava

resources/views/home.blade.php

```

1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5     <div class="col-md-8 col-md-offset-2">
6         <div class="panel panel-default">
7             <div class="panel-heading">Dashboard</div>
8
9             <div class="panel-body">
10                 @if (session('status'))
11                     <div class="alert alert-success" role="alert">
12                         {{ session('status') }}
13                     </div>
14                 @endif
15
16                 You are logged in!
17
18             </div>
19         </div>

```

```

20         </div>
21     </div>
22 @endsection

```

36.11 Muokataan /login-sivun näkymä

Muokataan **login**-sivun näkymä seuraavasti. Muutokset ovat pieniä, mutta muokkaaminen on helpointa korvaamalla koko sisältö

resources/views/auth/login.blade.php

```

1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container">
5          <div class="row">
6              <div class="col-md-8 col-md-offset-2">
7                  <div class="panel panel-default">
8                      <div class="panel-heading">{{ __('Login') }}</div>
9
10                     <div class="panel-body">
11                         <form method="POST" action="{{ route('login') }}"
12                             @csrf
13
14                         <div class="form-group row">
15                             <label for="email" class="col-md-4 col-
16
17                             <div class="col-md-6">
18                                 <input id="email" type="email" clas
19
20                                 @error('email')
21                                     <span class="invalid-feedback"
22                                         <strong>{{ $message }}</str
23                                     </span>
24                                 @enderror
25                             </div>
26                         </div>
27
28                         <div class="form-group row">
29                             <label for="password" class="col-md-4 c
30
31                             <div class="col-md-6">
32                                 <input id="password" type="password
33
34                                 @error('password')

```

```

35         <span class="invalid-feedback"
36             <strong>{{ $message }}</strong>
37         </span>
38         @enderror
39     </div>
40 </div>
41
42 <div class="form-group row">
43     <div class="col-md-6 col-md-offset-4">
44         <div class="form-check">
45             <input class="form-check-input"
46
47                 <label class="form-check-label"
48                     {{ __('Remember Me') }}
49                 </label>
50             </div>
51         </div>
52 </div>
53
54 <div class="form-group">
55     <div class="col-md-8 col-md-offset-4">
56         <button type="submit" class="btn btn-primary">
57             {{ __('Login') }}
58         </button>
59
60         @if (Route::has('password.request'))
61             <a class="btn btn-link" href="{{ route('password.request') }}">
62                 {{ __('Forgot Your Password') }}
63             </a>
64         @endif
65     </div>
66 </div>
67 </form>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 @endsection

```

36.12 Muokataan /register-sivun näkymä

Muokataan **register**-sivun näkymä seuraavasti. Muutokset ovat pieniä, mutta muokkaaminen on helpointa korvaamalla koko sisältö

```

1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container">
5      <div class="row">
6          <div class="col-md-8 col-md-offset-2">
7              <div class="panel panel-default">
8                  <div class="panel-heading">{{ __('Register') }}</div>
9
10                 <div class="panel-body">
11                     <form class="form-horizontal" method="POST" action="">
12                         @csrf
13
14                         <div class="form-group">
15                             <label for="name" class="col-md-4 control-label">Name</label>
16
17                             <div class="col-md-6">
18                                 <input id="name" type="text" class="form-control">
19
20                                 @error('name')
21                                     <span class="invalid-feedback">
22                                         <strong>{{ $message }}</strong>
23                                     </span>
24                                 @enderror
25                             </div>
26                         </div>
27
28                         <div class="form-group row">
29                             <label for="email" class="col-md-4 control-label">Email</label>
30
31                             <div class="col-md-6">
32                                 <input id="email" type="email" class="form-control">
33
34                                 @error('email')
35                                     <span class="invalid-feedback">
36                                         <strong>{{ $message }}</strong>
37                                     </span>
38                                 @enderror
39                             </div>
40                         </div>
41
42                         <div class="form-group row">
43                             <label for="password" class="col-md-4 control-label">Password</label>
44

```

```

45         <div class="col-md-6">
46             <input id="password" type="password"
47
48             @error('password')
49                 <span class="invalid-feedback"
50                     <strong>{{ $message }}</strong>
51                 </span>
52             @enderror
53         </div>
54     </div>
55
56     <div class="form-group">
57         <label for="password-confirm" class="col-
58
59         <div class="col-md-6">
60             <input id="password-confirm" type="
61         </div>
62     </div>
63
64     <div class="form-group">
65         <div class="col-md-6 col-md-offset-4">
66             <button type="submit" class="btn btn-
67                 {{ __('Register') }}
68             </button>
69         </div>
70     </div>
71 </form>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 @endsection

```

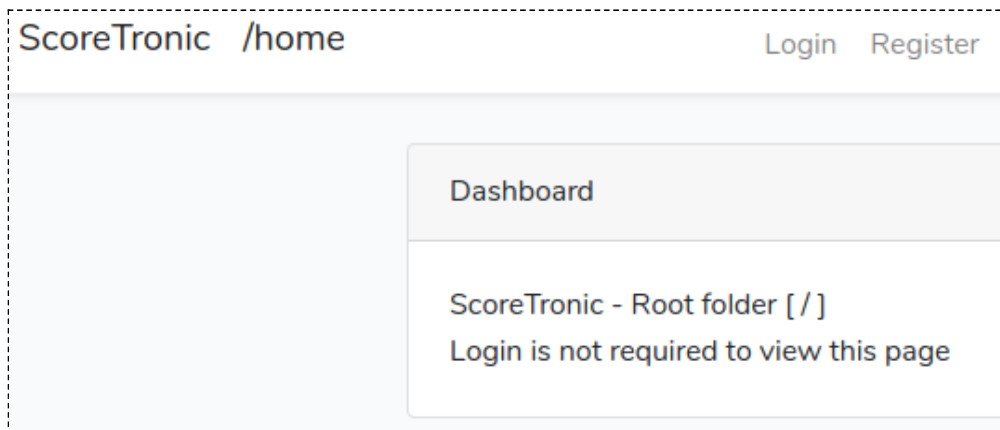
36.13 Yhteenveto näkymistä

Luodussa uudessa sovelluksessa on näkymät

Reitti	Selitys
/	Ei vaadi kirjautumista
/home	vaatii kirjautumisen, ohjautuu login-sivulle, jos käyttäjä ei ole kirjautunut
/login	kirjautumislomake
/register	rekisteröitymislomake

Näkymät ovat suunnilleen seuraavan näköisiä

Päänäkymä



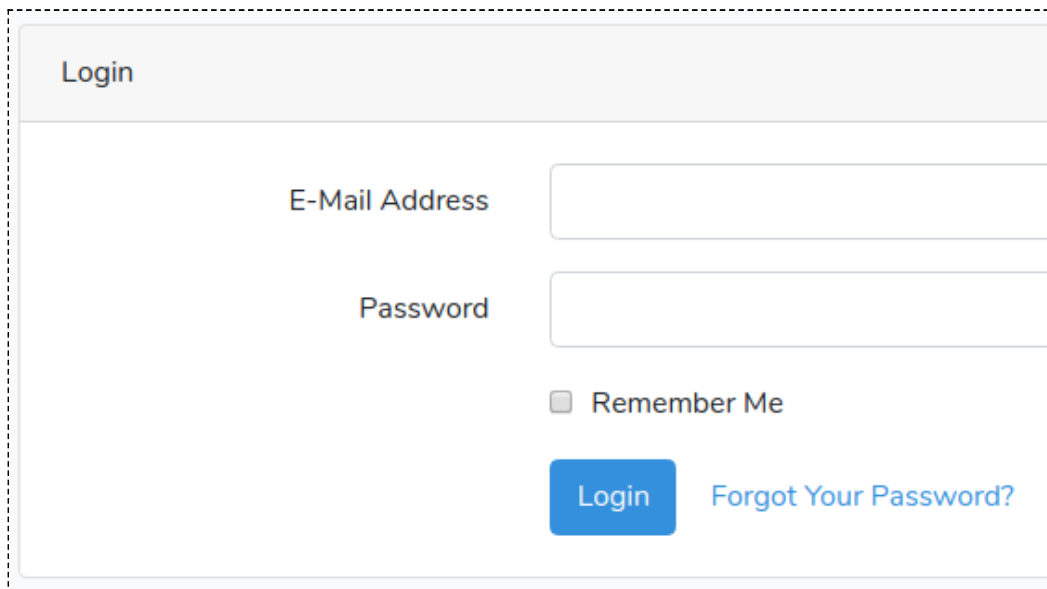
ScoreTronic /home Login Register

Dashboard

ScoreTronic - Root folder [/]

Login is not required to view this page

Login-näkymä



Login

E-Mail Address

Password

☐ Remember Me

Login Forgot Your Password?

36.14 Käyttäjien rekisteröinti

Rekisteröidy sivulla <http://192.168.1.126/~testi/projekti01/public/register> ja syötä tiedot

--	--
Name	Guru Ken
Email	G1234@student.jamk.fi
Password	sala1234

Huomaat, että pääsit kirjautumaan osoitteeseen
<http://192.168.1.126/~testi/projekti01/public/home>

Katso tietokannasta, että käyttäjätunnus on luotu:

```
mysql> select * from users;
```

```
+----+-----+-----+-----+
| id | name      | email                | email_verified_at |
+----+-----+-----+-----+
|  1 | Guru Ken  | G1234@student.jamk.fi | NULL              |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

Rekisteröi vielä tunnukset

--	--
Name	Alainen Kim
Email	A1234@student.jamk.fi
Password	sala1234
--	--
Name	Admin Admin
Email	admin@student.jamk.fi
Password	sala1234

Tietokannan **laraveldb** `users`-taulun sisältö tulisi olla näiden lisäysten jälkeen seuraava

```
mysql> select * from users;
```

```
+----+-----+-----+-----+
| id | name      | email                | email_verified_at |
+----+-----+-----+-----+
|  1 | Guru Ken  | G1234@student.jamk.fi | NULL              |
|  2 | Alainen Kim | A1234@student.jamk.fi | NULL              |
|  3 | Admin Admin | admin@student.jamk.fi | NULL              |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Huomaa erityisesti käyttäjien id-numerot. Niihin tulee ainakin kiinnittää huomiota jatkossa.

36.15 HomeControllerin autorisointi

Tämä kohta **ei vaadi** toimenpiteitä, ainoastaan huomioinnin

Huomaa, että `app/Http/Controllers/HomeController.php`-tiedostossa on

```
1 public function __construct()
2 {
3     $this->middleware('auth');
4 }
```


Se tarkoittaa sitä, että kaikki HomeControllerin kautta kiertävät pyynnot onnistuvat vain kirjautuneilta käyttäjiltä

36.16 Logoutin redirect

Tämä kohta **ei vaadi** toimenpiteitä, ainoastaan huomioinnin

- Huomaa, että logoutin jälkeen käyttäjä ohjataan kirjautumattomana näkymään <http://192.168.1.126/~testi/projekti01/public/>
- Käyttäjä voitaisiin haluttaessa ohjata logoutin jälkeen esim. näkymään <http://192.168.1.126/~testi/projekti01/public/home> seuraavasti:

Lisäisit tiedoston `app/Http/Controllers/Auth/LoginController.php` alkuun rivin:

```
1 | ...  
2 | use Illuminate\Http\Request;  
3 | ...
```

ja lisäisit samaan tiedostoon myös metodin

```
1 | public function logout(Request $request){  
2 |     $this->guard()->logout();  
3 |     $request->session()->flush();  
4 |     $request->session()->regenerate();  
5 |  
6 |     return redirect()->route('home');  
7 | }
```

36.17 Tietokannan populointi

Populoidaan tietokanta eli lisätään tietokantaan esimerkkitietoa.

Tarkista kohdan 36.11 mukaan ensin käyttäjien id-numerot. Huomaa, nämä voivat vaihdella sinun järjestelmässä. Tässä käyttäjien id:t ovat siis

id	Nimi
1	Guru Ken
2	Alainen Kim
3	Admin Admin

Lisätään kaksi harjoitussarjaa ja kumpaankin kaksi tehtävää. Anna seuraavat SQL-lauseet mysql-komentokehotteessa

```

1  INSERT INTO hsarja (nro, palautuspvm, kuvaus) VALUES
2  (1, '2020-11-11', 'Harjoitukset 1'),
3  (2, '2020-11-11', 'Harjoitukset 2');
4
5  INSERT INTO tspec (hnro, tnro, maxpist) VALUES
6  (1, 1, 2),
7  (1, 2, 4),
8
9  (2, 1, 4),
10 (2, 2, 8);

```

Katsotaan lopputulos

```

mysql> select * from hsarja;
+-----+-----+-----+
| nro | palautuspvm | kuvaus |
+-----+-----+-----+
| 1 | 2020-11-11 | Harjoitukset 1 |
| 2 | 2020-11-11 | Harjoitukset 2 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from tspec;
+-----+-----+-----+-----+
| hnro | tnro | maxpist | kuvaus |
+-----+-----+-----+-----+
| 1 | 1 | 2 | NULL |
| 1 | 2 | 4 | NULL |
| 2 | 1 | 4 | NULL |
| 2 | 2 | 8 | NULL |
+-----+-----+-----+-----+
4 rows in set (0.00 se

```

Lisätään käyttäjälle Ken Guru (id=1) tehtäväpalautukset opettajan ok/ek-kommenteineen. Opettaja on hyväksynyt kaikki muut täysin pistein, mutta viimeisestä tehtävästä opettaja on antanut 0p vaikka opiskelija on ehdottanut 8/8. Palautettujen harjoitusten palautusosoitteena käytetään urlia

<http://student.labranet.jamk.fi/~ara/ttmstest>

Anna seuraavat SQL-lauseet mysql-komentokehoteessa:

```

1  INSERT INTO tpalautus (student_id, hnro, tnro, initpoints, finalpoints,
2  (1, 1, 1, 2, 2, 'ok', 'http://student.labranet.jamk.fi/~ara/ttmstes
3  (1, 1, 2, 4, 4, 'ok', 'http://student.labranet.jamk.fi/~ara/ttmstes
4  (1, 2, 1, 4, 4, 'ok', 'http://student.labranet.jamk.fi/~ara/ttmstes
5  (1, 2, 2, 8, 0, 'ek', 'http://student.labranet.jamk.fi/~ara/ttmstes

```

Katsotaan kaikki palautukset, anna seuraava SQL-lause mysql-komentokehoteessa:

```
1 SELECT users.id, users.name,
2     tpalautus.hnro as h, tpalautus.tnro as t, tpalautus.palautusaika as a,
3     tpalautus.initpoints as `in`, tpalautus.finalpoints as `fi`,
4     tspec.maxpist as maks,
5     hsarja.kuvaus
6 FROM users
7 INNER JOIN tpalautus
8     ON users.id = tpalautus.student_id
9 INNER JOIN tspec
10    ON tspec.hnro = tpalautus.hnro AND tspec.tnro = tpalautus.tnro
11 INNER JOIN hsarja
12    ON hsarja.nro = tspec.hnro;
```

tuloksena tulisi olla esim.

id	name	h	t	palautusaika	in	fi	cmt
1	Guru Ken	1	1	2019-11-02 07:57:32	2	2	ok
1	Guru Ken	1	2	2019-11-02 07:57:32	4	4	ok
1	Guru Ken	2	1	2019-11-02 07:57:32	4	4	ok
1	Guru Ken	2	2	2019-11-02 07:57:32	8	0	ek

4 rows in set (0.00 sec)

36.18 Näkymä opiskelijalistalle

36.18.1 Tehdään aluksi reitti reittitiedostoon

routes/web.php

```
1 ...
2 Route::get('users', 'UserController@list_all');
```

36.18.2 Luodaan UserController

```
php artisan make:controller UserController
```

ja muokataan kontrolleritiedostoa seuraavasti

app/Http/Controllers/UserController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\User;
6  use Illuminate\Http\Request;
7
8  class UserController extends Controller
9  {
10     public function list_all()
11     {
12         $users = User::All();
13         return view('users')->with('users', $users);
14     }
15 }
```

36.18.3 Luodaan users.blade.php-näkymä

resources/views/users.blade.php

```
1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container">
5      <div class="row">
6          <div class="col-md-8 col-md-offset-2">
7              <div class="panel panel-default">
8                  <div class="panel-heading">Kaikki opiskelijaresurssit
9  @if (session('info'))
10     <div class="alert alert-info">
11         {{ session('info') }}
12     </div>
13 @endif
14
15         <div class="panel-body">
16     <table class="table table-striped">
17         <thead>
18             <tr>
19                 <th scope="col">id</th>
20                 <th scope="col">name</th>
21                 <th scope="col">email</th>
22                 <th scope="col">luotu</th>
23             </tr>
24         </thead>
25         <tbody>
```

```

26
27 @foreach ($users as $opiskelija)
28
29     <tr class="bg-success">
30         <th scope="row">{{ $opiskelija->id }}</th>
31         <td>{{ $opiskelija->name }}</td>
32         <td>{{ $opiskelija->email }}</td>
33         <td>{{ $opiskelija->created_at }}</td>
34         <td><a href="/scores?id={{ $opiskelija->id }}"> Harkkapalautu
35     </tr>
36
37 @endforeach
38
39 </tbody>
40 </table>
41
42     </div>
43
44 </div>
45 </div>
46 @endsection

```

36.18.4 Tutkitaan pudotusvalikkoo

resources/views/layouts/app.blade.php

Huomataan, että pudotusvalikko on toteutettu seuraavalla HTML-merkkauksella

```

1 <!-- Left Side Of Navbar -->
2 <ul class="nav navbar-nav">
3
4 <li class="dropdown">
5     <a class="dropdown-toggle" data-toggle="dropdown" href="#"
6 <ul class="dropdown-menu" aria-labelledby="download">
7 <li><a href="{{ url('/users') }}">Opiskelijalista</a></li>
8 <li class="divider"></li>
9 <li><a href="{{ url('/harkat') }}">Harkkapalautukset</a></li>
10 <li class="divider"></li>
11 <li><a href="{{ url('/toiminto22') }}">Toiminto 22</a></li>
12 <li class="divider"></li>
13 <li><a href="{{ url('/toiminto33') }}">Toiminto 33</a></li>
14
15 </ul>
16 </li>
17 </ul>

```

Testaa uuden pudotusvalikon toiminta. Muut lisätyt valikkokohdat eivät vielä toimi. Niiden uudet toiminnot tullaan muokkaamaan jatkossa tähän pudotusvaikkoon. Käyttöliittymältään users-näkymän eli opiskelijalistan tulisi näyttää suunnilleen oheiselta

ScoreTronic /home Demotoiminnot ▾

Opiskelijalista 📄			
Harkkapalautukset			
Toiminto 22			
Toiminto 33			
		email	luotu
		G1234@student.jamk.fi	2019-11-
		A1234@student.jamk.fi	2019-11-
3	Admin Admin	admin@student.jamk.fi	2019-11-