

## MODUL 13 Web Framework

---

### 13.1. Deskripsi Singkat

Web framework adalah suatu software yang dirancang untuk mendukung pengembangan suatu aplikasi web. Pada web framework disediakan suatu standar mekanisme utk membangun (*build*) dan memasang (*deploy*) aplikasi web di internet. Fitur-fitur yang biasanya tersedia diantaranya, library untuk mengakses database, templating framework, manajemen session, security, dan lain-lain. Web framework biasanya mempromosikan code reuse untuk memudahkan proses pengembangan aplikasi web.

Secara umum web framework application terbagi menjadi frontend framework dan backend framework. Front end framework fokus pada elemen visual website atau aplikasi di sisi client yang berinteraksi dengan user, meliputi CSS dan Javascript framework. Sedangkan backend framework fokus pada sisi server, yaitu sisi website yg tidak terlihat oleh user, diantaranya PHP framework.

### 13.2. Tujuan Praktikum

Setelah menyelesaikan praktikum pada modul ini, mahasiswa diharapkan dapat mengenal bagaimana menggunakan web framework application untuk mempermudah pengembangan aplikasi web.

### 13.3. Material Praktikum

Praktikum menggunakan editor teks seperti Notepad atau Notepad++ atau aplikasi editor teks lainnya yang tersedia untuk menulis HTML dan program PHP. Pada praktikum ini juga menggunakan web server Apache dan program PHP.

### 13.4. Kegiatan Praktikum

#### 13.4.1. Front-end framework: Bootstrap

Bootstrap merupakan salah satu front-end framework yang sering digunakan. Saat modul ini dibuat, versi terbaru dari Bootstrap adalah Bootstrap 5.2. Untuk melihat dokumentasi cara memulai membuat aplikasi web dengan Bootstrap 5.2, silakan kunjungi halaman berikut.

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Selanjutnya, Anda akan membuat halaman web menggunakan Bootstrap.

1. Buat folder project web dengan nama `webbs`. Kemudian buat file `index.html` dengan kode berikut.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Bootstrap Demo</title>
  </head>
  <body>
  </body>
</html>
```

2. Kemudian sisipkan bootstrap CSS pada tag `head` dan bootstrap JS pada akhir tag `body`.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFB6D0itfPri4tjfhXaWutUpFmBp4vmVor"
crossorigin="anonymous">
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>
```

Selain menggunakan link CSS dan JS eksternal, dapat juga menggunakan link file direktori di server dengan menyimpan file CSS dan JS ke folder project web.

3. Kemudian buat layout halaman web yang terdiri dari header dan 3 kolom artikel dengan menambahkan kode berikut di tag `body`.

```
<div class="container-fluid p-5 bg-primary text-white text-
center">
  <h1>Halaman Web dengan Bootstrap</h1>
  <p>Resize halaman responsive ini untuk melihat efeknya!</p>
</div>

<div class="container mt-5">
  <div class="row">
    <div class="col-sm-4">
      <h3>Kolom 1</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...</p>
    </div>
    <div class="col-sm-4">
```

```

        <h3>Kolom 2</h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit...</p>
        <p>Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...</p>
    </div>
    <div class="col-sm-4">
        <h3>Kolom 3</h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit...</p>
        <p>Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...</p>
    </div>
</div>
</div>

```

4. Akses halaman web tersebut melalui browser dan resize untuk melihat efek responsifnya. Baca dokumentasi mengenai layout container pada halaman berikut:

<https://getbootstrap.com/docs/5.2/layout/containers/>

5. Kemudian ubah warna header menjadi warna orange gradient dengan text berwarna hitam. Baca dokumentasi mengenai background pada halaman berikut:

<https://getbootstrap.com/docs/5.2/utilities/background/>

6. Selanjutnya, Anda dapat melihat contoh kode untuk halaman-halaman yang dirancang menggunakan bootstrap pada halaman berikut:

<https://getbootstrap.com/docs/5.2/examples/>

Download source code untuk halaman sign-in dan dashboard. Lalu buat folder project webbs\_dashboard dan kombinasikan kode pada dua halaman tersebut sehingga Anda memiliki dua halaman html yaitu index.html untuk halaman login dan dashboard.html untuk halaman dashboard.

7. Tambahkan kode pada tombol Sign in dan Sign out. Jika Anda menekan tombol Sign in, maka akan berpindah ke halaman dashboard (dashboard.html) dan jika Anda menekan tombol Sign out, maka akan berpindah kembali ke halaman Sign in (index.html).

### 13.4.2. Front-end framework: Vue.js

Vue.js merupakan salah satu front-end framework berbasis Javascript untuk membuat tampilan interface website yang interaktif. Selanjutnya, ikuti langkah-langkah berikut untuk membuat project web menggunakan Vue.js.

1. Sebelum menginstal Vue.js, instal terlebih dahulu Node.js dan npm yang dapat diperoleh dari link berikut.

<https://nodejs.org/en/download/>

Kemudian cek apakah node.js dan npm sudah berhasil terinstal atau tidak dengan mengetikkan perintah berikut di command prompt.

```
C:\Users\user>node -v
v16.15.0

C:\Users\user>npm -v
8.5.5
```

Jika sudah terinstal, maka akan ditampilkan versi dari node.js dan npm-nya.

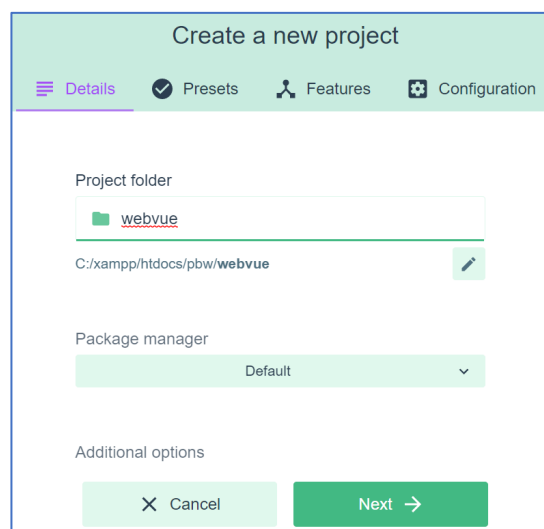
2. Anda dapat membuat project web menggunakan Vue dengan menggunakan Vue CLI. Vue CLI adalah standard tooling untuk pengembangan web Vue.js yang menyediakan user interface untuk memudahkan konfigurasi pembuatan project web dengan Vue. Vue CLI dapat diinstal dengan mengetikkan perintah berikut di command prompt.

```
npm install -g @vue/cli
```

Untuk menjalankan vue-cli, ketikkan perintah berikut.

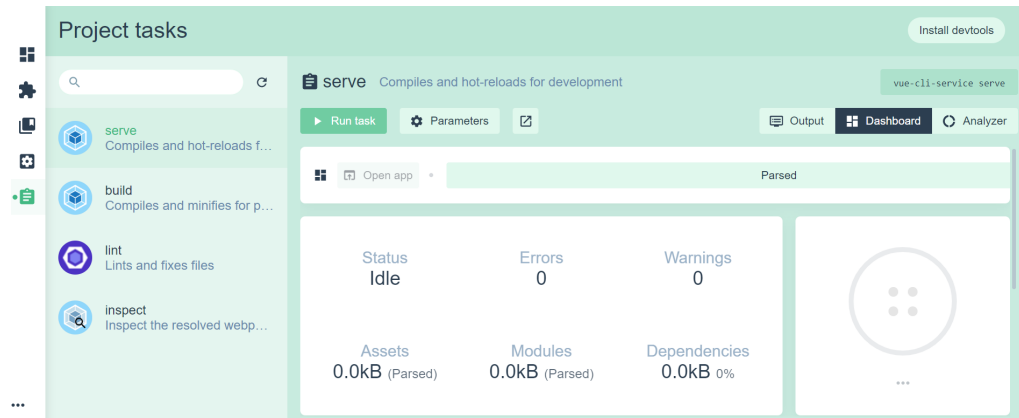
```
vue ui
```

Kemudian halaman web berikut akan otomatis terbuka dengan alamat <http://localhost:8000/>



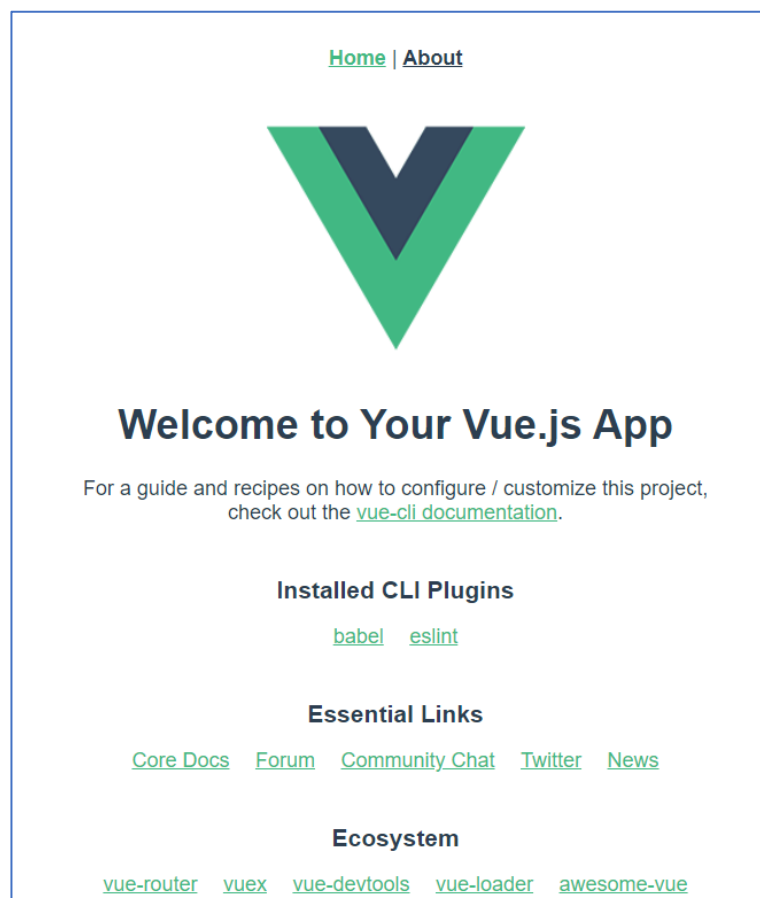
Kemudian buat project dengan nama `webvue` lalu klik tombol `Next`.

3. Pada halaman manajemen project `webvue`, buka tab `Plugins`, lalu instal plugin `vue-router`.
4. Setelah plugins yang dibutuhkan terinstal, jalankan project web `webvue` dengan cara membuka tab `Tasks`, klik `serve`, dan klik tombol `Run Task`.

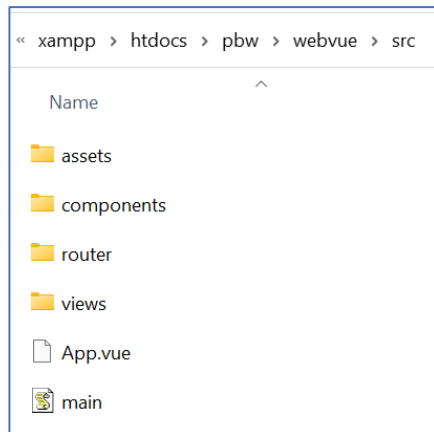


Klik tombol `Open app` untuk melihat tampilan halaman web default yang telah dibuat. Halaman yang tampil yaitu sebagai berikut.

<http://localhost:8080/>



5. Kemudian, buka folder project web Anda (folder bernama `webvue`) menggunakan File Explorer. Lalu buka folder `src`. Terdapat beberapa file dan folder sebagai berikut.

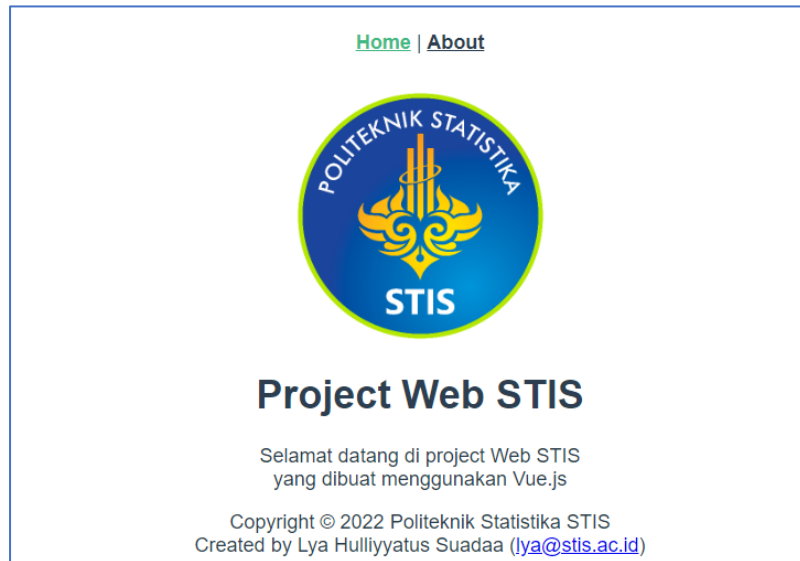


Buka terlebih dahulu `main.js` menggunakan text editor. Anda akan melihat bahwa terdapat kode memanggil `App.vue`. Selanjutnya buka file `App.vue`, perhatikan tag `template` berikut.

```
<template>
  <nav>
    <router-link to="/">Home</router-link> |
    <router-link to="/about">About</router-link>
  </nav>
  <router-view/>
</template>
```

Pada kode di atas, terdapat informasi menu navigasi yang sudah dibuat, yaitu `Home` dan `About`. Untuk mendapatkan informasi mengenai pengaturan router dan komponen menu tersebut, buka folder `router` dan baca file `index.js`. Perhatikan isi dari konstanta `routes` yang merupakan associative array dengan key `path`, `name`, dan `component`. Anda dapat melihat nama view dari setiap menu pada value dari `component`. Secara fisik, file tampilan halaman tersebut dapat ditemukan di folder `views`.

6. Ubah halaman `Home` dengan mengubah isi dari file `HomeView.vue` pada folder `views` sehingga tampilan `Home` yang baru menjadi seperti gambar berikut.



Hint: Hapus kode selain tag `template` dan isinya pada `HomeView.vue`, kemudian tulis kode HTML yang sesuai. Ubah footer dengan nama dan alamat email masing.

#### 13.4.3. Back-end framework: CodeIgniter

CodeIgniter merupakan salah satu back-end framework berbasis PHP. Selanjutnya, ikuti langkah-langkah berikut untuk membuat project web menggunakan CodeIgniter.

1. Terdapat beberapa cara yang dapat dipilih untuk menginstal CodeIgniter. Silakan membaca lebih detail cara instalasinya pada halaman berikut.

[https://codeigniter.com/user\\_guide/installation/](https://codeigniter.com/user_guide/installation/)

Pada praktikum ini, Anda akan mencoba melakukan instalasi dengan menggunakan Composer. Silakan baca dokumentasinya untuk mengetahui kelebihan instalasi dengan Composer. Jika Anda belum pernah menginstal Composer, kunjungi halaman berikut untuk mengunduh file instalasi Composer dan lakukan instalasi.

<https://getcomposer.org/download/>

Kemudian buka command prompt, pindahkan path ke direktori `htdocs`, dan jalankan perintah berikut.

```
composer create-project codeigniter4/appstarter webci --no-dev
```

Kemudian buka direktori tempat Anda membuat project tersebut. Pastikan sudah ada folder baru bernama `webci`. Di dalam folder tersebut terdapat

folder dan file yang dibutuhkan untuk membuat project web dengan CodeIgniter.

2. Akses web tersebut dari browser. Jika Anda menggunakan Apache sebagai server, maka halaman web tersebut dapat dikunjungi dengan menambahkan "public" di akhir link url, contohnya:

```
http://localhost/webci/public/
```

Karena CodeIgniter sudah memiliki internal server yang dapat digunakan untuk mempermudah menjalankan halaman web pada tahap development, maka selanjutnya kita akan menggunakan internal server ini. Untuk menjalankan internal server di CodeIgniter, ketikkan perintah berikut di command prompt.

```
php spark serve
```

Selanjutnya, project `webci` dapat diakses melalui link berikut.

```
http://localhost:8080/
```

Jika internal server gagal dijalankan, lakukan langkah-langkah berikut.

- a. Buka `php.ini`, aktifkan `extension=intl` dengan menghapus tanda `;"` di depannya, lalu simpan kembali file `php.ini`.

```
extension=intl
```

- b. Jalankan perintah berikut di command prompt

```
composer install
```

3. Perhatikan file dan folder yang ada di dalam folder project `webci`. Di dalam folder tersebut terdapat file bernama `env`. Duplicate file tersebut, dan rename file barunya menjadi `.env`. Lalu buka file `.env`, dan ubah baris berikut:

```
# CI_ENVIRONMENT = production
```

menjadi:

```
CI_ENVIRONMENT = development
```

Reload project `webci` di browser, dan perhatikan apakah ada perubahan pada footernya? Baca dokumentasi mengenai hal ini pada link berikut.

```
https://codeigniter.com/user\_guide/general/configuration.html  
https://codeigniter.com/user\_guide/general/environments.html
```



4. Buka kembali file `.env`, dan ubah baris berikut:

```
# app.baseURL = ''
```

menjadi:

```
app.baseURL = 'http://localhost:8080/'
```

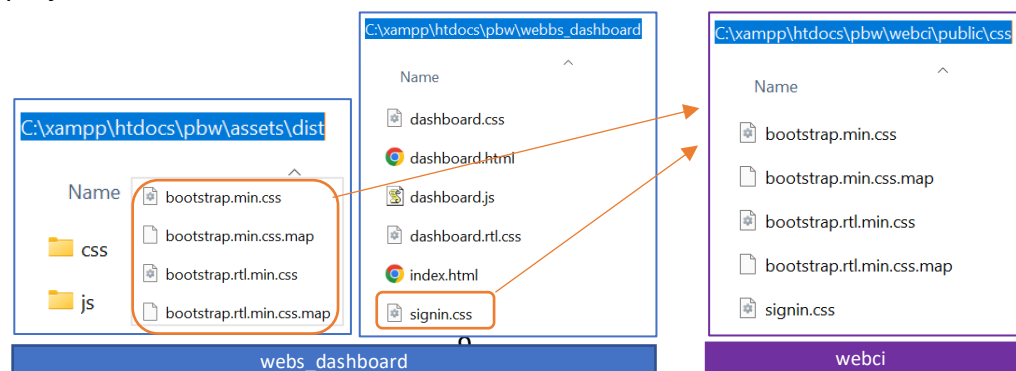
Isi dari `baseURL` disesuaikan dengan link server yang dijalankan pada setiap environment. Karena pada saat ini Anda sedang menggunakan internal server Codelgniter, maka gunakan `baseURL 'http://localhost:8080/'`.

5. Folder inti dari project Codelgniter yaitu folder `app`. Perhatikan isi dari folder `app` tersebut. Folder `Models`, `Views`, dan `Controllers` di dalam folder `app` digunakan untuk menerapkan MVC pattern, yaitu dengan memisahkan kode untuk model, view, dan controller. Silakan buka file `Home.php` pada folder `Controllers` dan perhatikan isi file tersebut, terdapat fungsi `index` sebagai berikut.

```
public function index()
{
    return view('welcome_message');
}
```

View atau halaman yang dipanggil pada fungsi `index` tersebut bernama `welcome_message`. Buka file `welcome_message` pada folder `Views` dan perhatikan isinya.

6. Selanjutnya Anda akan mengganti view `welcome_message` pada fungsi `index` dengan halaman sign-in. Copy file `index.html` yang merupakan halaman html dari form sign-in menggunakan bootstrap dari project `web_dashboard`, paste ke folder `Views` pada project `webci`, ganti ekstensi filenya dari `index.html` menjadi `index.php`. Kemudian, copy juga semua file `css` yang digunakan oleh halaman tersebut (satu folder `css` di folder `assets` dan file `sign-in.css`). Paste ke folder `public` pada project `webci`. Satukan dalam folder `css`.

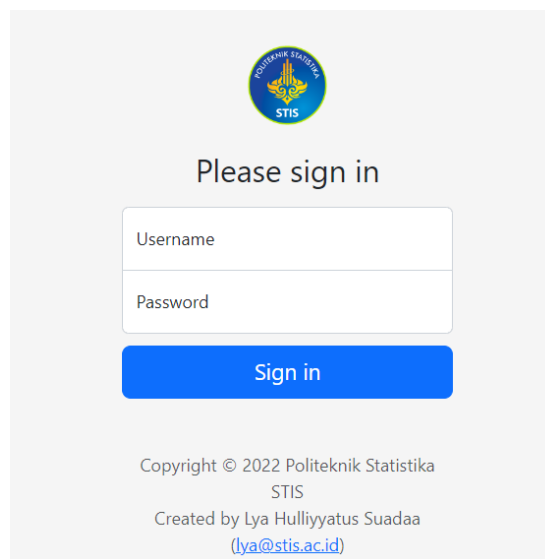


Kemudian buka file `index.php` di folder `Views`, ubah referensi pada tag link css menjadi:

```
<link href="/css/bootstrap.min.css" rel="stylesheet">
<link href="/css/signin.css" rel="stylesheet">
```

Buka kembali file `Home.php` pada folder `Controllers` dan ubah view pada fungsi `index` dari `"welcome_message"` menjadi `"index"`.

Kemudian akses `webci` dari browser, pastikan yang tampil adalah halaman sign-in. Modifikasi tampilannya sehingga menjadi seperti gambar berikut.



File resources yang dapat diakses publik seperti image, css, dan js dapat disimpan di folder `public`.

7. Selanjutnya Anda akan membuat model untuk mengakses tabel `user` yang Anda sudah buat pada praktikum sebelumnya. Ubah terlebih dahulu konfigurasi database pada `webci` dengan mengubah baris berikut di file `.env`

```
# database.default.hostname = localhost
# database.default.database = ci4
# database.default.username = root
# database.default.password = root
# database.default.DBDriver = MySQLi
```

menjadi:

```
database.default.hostname = localhost
database.default.database = <nama_database>
database.default.username = <username>
database.default.password = <password>
```

```
database.default.DBDriver = MySQLi
```

Sesuaikan isiannya dengan nama database, username, dan password yang Anda gunakan.

Kemudian buat file `UserModel.php` pada folder `Models` dengan isi kode sebagai berikut.

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class UserModel extends Model
{
    protected $table      = 'user';
    protected $primaryKey = 'username';
}
```

8. Kemudian buat fungsi untuk sign in di Controller. Tambahkan kode fungsi berikut di `Home.php` pada folder `Controllers`.

```
public function signinAuth()
{
    $session = session();
    $userModel = new UserModel();
    $username = $this->request->getVar('username');
    $password = $this->request->getVar('password');

    $data = $userModel->where('username', $username)->first();

    if($data){
        $pass = $data['password'];
        if($password=== $pass){
            $ses_data = [
                'username' => $data['username'],
                'isSignedIn' => TRUE
            ];
            $session->set($ses_data);
            return redirect()->to('/Home/meeting');
        }else{
            $session->setFlashdata('msg', 'Password is
incorrect. ');
            return redirect()->to('/Home');
        }
    }else{$session->setFlashdata('msg', 'Username does not
exist. ');

    return redirect()->to('/Home');
```

```
}
}
```

Jangan lupa tambahkan kode berikut sebelum pendefinisian Class pada `Home.php` agar dapat mengakses `UserModel.php` tanpa menuliskan lokasinya.

```
use App\Models\UserModel;
```

9. Buat halaman view bernama `meeting.php` yang akan ditampilkan setelah user berhasil sign in.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <meta name="description" content="">
    <link href="/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <header class="navbar navbar-dark sticky-top bg-dark flex-md-
nowrap p-0 shadow">
      <div class="navbar-nav">
        <div class="nav-item text-nowrap">
          <a class="nav-link px-3" href="Signout">Sign out</a>
        </div>
      </div>
    </header>
    <div class="container-fluid">
      Hello, <?=$username;?>
      <h1>Data Meeting</h1>
    </div>
  </body>
</html>
```

Lalu buat controller untuk menampilkan view tersebut. Tambahkan fungsi `meeting` pada controller `Home.php`.

```
public function meeting()
{
    $session = session();
    $data = [
        'username' => $session->get('username')
    ];
    return view('meeting', $data);
}
```

10. Tambahkan pula fungsi `signout` pada controller `Home.php` yang berisi perintah untuk menghapus session dan kembali ke halaman sign in.

```
public function signout()
{
    $session = session();
    $session->destroy();
    return redirect()->to('/Home');
}
```

11. Akses kembali project `webci` melalui browser. Pastikan halaman `sign in` sudah berfungsi dengan baik, menampilkan halaman `meeting` setelah berhasil `sign in`, dan kembali ke halaman `sign in` setelah menekan `sign out`. Pastikan pula `session` terbuat saat `sign in` dan terhapus saat `signout`.

### 13.5. Responsi

Kerjakan sesuai dengan yang dijelaskan pada bagian Kegiatan Praktikum. Simpan tangkapan layar hasil pekerjaan Anda untuk empat project (`webbs`, `webbs_dashboard`, `webvue`, `webci`) dari ketiga kegiatan praktikum dalam file docx. Simpan ulang file tersebut dalam format pdf, dan beri nama dengan format `<<nim>>_modul13.pdf`, contoh: `192191234_modul13.pdf`. Kumpulkan file tersebut sebagai responsi melalui Google Classroom.