

```
<?php

namespace App\Controllers;

use App\Models\M_model;
use App\Models\PostModel;
use App\Models\CommentModel;
use Intervention\Image\ManagerInterfaceStatic as Image;
```

Berikut merupakan model yang saya gunakan yakni M_model untuk menyimpan beberapa perintah dasar selanjutnya ada PostModel dan CommentModel yang saya gunakan untuk menggunakan Fitur comment dalam aplikasi Gallery selanjutnya saya menggunakan fitur composer yang berisi Intervention image untuk melakukan auto resize image

```
public function index()
{
    echo view('header');
    // echo view('/admin/menuut
    echo view('/admin/login');
    // echo view('footer');
}
```

```

public function aksi_login()
{
    $u = $this->request->getPost('username');
    $p = $this->request->getPost('password');
    $model = new M_model();
    $data = array(
        'username' => $u,
        'password' => md5($p)
    );
    $cek = $model->getWhere2('user', $data);

    if ($cek > 0) {
        session()->set('id_user', $cek['id_user']);
        session()->set('username', $cek['username']);
        // session()->set('email', $cek['email']);
        session()->set('level', $cek['level']);
        return redirect()->to('/home/dashboard');
    } else {
        // Tambahkan kode berikut
        session()->setFlashdata('error', 'Salah password');
        return redirect()->to('/home/index');
    }
}

```

untuk function pertama yaitu index berisi halaman login page dan terdapat aksi_login untuk mengeksekusi login jadi seorang user harus / wajib melakukan login jika ingin mengakses website gallery terdapat username dan password seorang user harus memasukkan data yang tepat jika tidak maka akan kembali redirect ke login page

berikut merupakan tampilan login

Group 1 Fotogeny | Photography Agency X

localhost:8080/Home

Login

Username *

Admin

Password *

•

☐ Remember me [Login](#) [Register](#)

```
public function like($id){  
    $model = new M_Model();  
    // $where2=array('id_post'=>$id);  
    $data2 = array(  
        'post' => $id,  
        'createdBy' => session()->get('id_user'),  
        'created_at' => date('Y-m-d H:i:s')  
    );  
  
    // print_r($data2);  
    $model->simpan('like', $data2);  
  
    return redirect()->to ('/Home/portofolio/'.$id);  
}
```

```

public function dislike($id){
    $model = new M_Model();
    // $where2=array('id_post'=>$id);
    $data2 = array(
        'post' => $id,
        'createdBy' => session()->get('id_user')
    );

    // print_r($data2);
    $model->hapus('like', $data2);

    return redirect()->to ('/Home/portofolio/'.$id);
}

```

selanjutnya terdapat function like dan dislike

```

public function aksi_like($id)
{
    if(session()->get('level')==1) {
        $model = new M_galeri();

        $idUser = session()->get('id_user');

        // Periksa apakah user sudah memberikan like atau belum
        if (!$model->isLiked($id, $idUser)) {
            // Jika belum, lakukan like
            $data1 = array(
                'gambar' => $id,
                'user' => $idUser
            );
            $model->simpan('like', $data1);
        } else {
            // Jika sudah, lakukan unlike
            $model->hapusLike($id, $idUser);
        }

        return redirect()->to('home/portofolio/'.$id);
    } else {
        return redirect()->to('login');
    }
}

```

kemudian terdapat aksi_like untuk mengeksekusi like

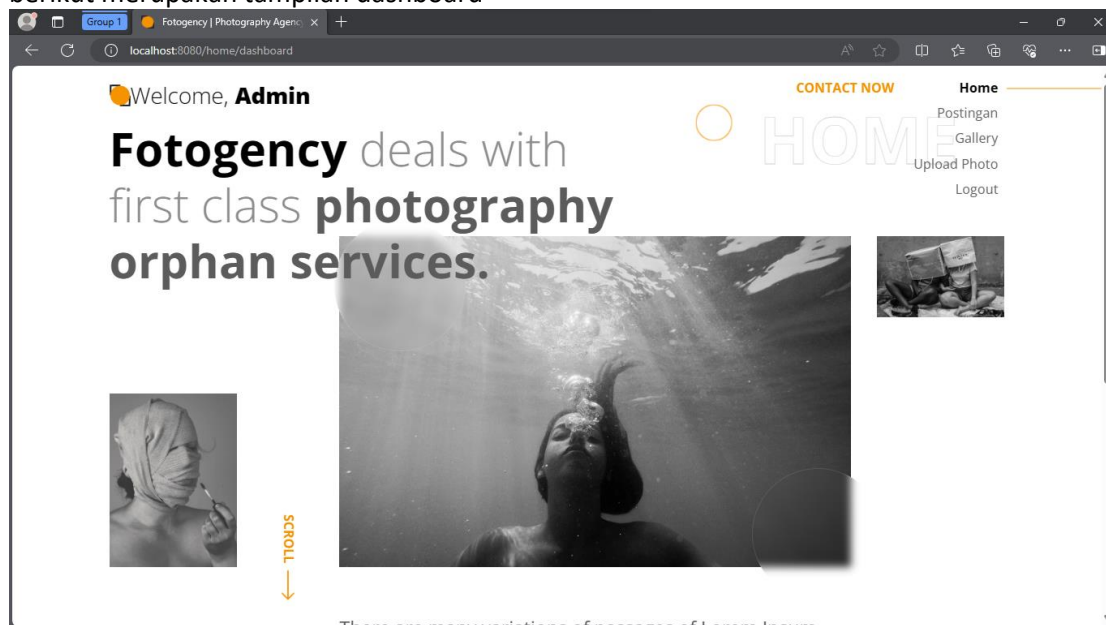
```

public function dashboard()
{
    if (session()->get('level')==1) {
        echo view('header');
        // echo view('menu');
        echo view('dashboard');
        echo view('footer');
    }
    else {
        return redirect()->to('/home');
    }
}

```

selanjutnya terdapat halaman dashboard / halaman utama saat pertama kali login (Jika user melakukan login maka akan di direct ke halaman dashboard)

berikut merupakan tampilan dashboard



```

public function gallery()
{
    $model = new M_model();
    $id_user = session()->get('id_user');

    // Fetch photos based on user_id directly from the 'photos' table
    $data['galeri'] = $model->where('user_id', $id_user)->findAll();

    echo view('header');
    echo view('gallery', $data);
    echo view('footer');
}

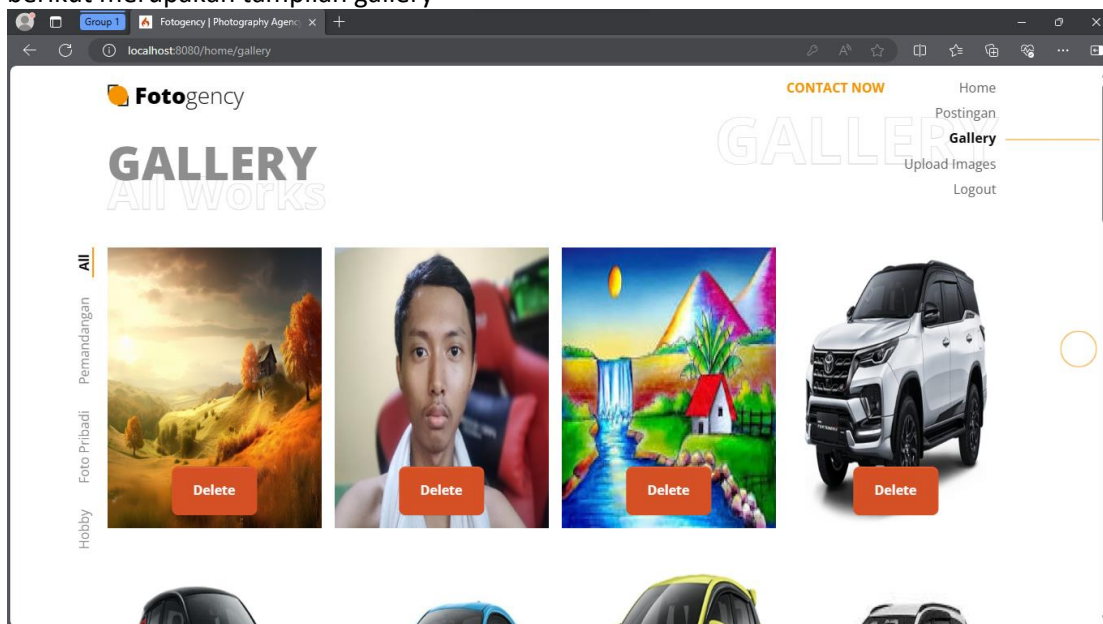
```

Selanjutnya terdapat function gallery yang digunakan untuk menuju ke view gallery yang berisi foto foto yang di input/upload oleh user dan yang saya gunakan ialah

\$id_user = session()->get('id_user');

\$data['galeri'] = \$model->where('user_id', \$id_user)->findAll();

yang bertujuan untuk menampilkan foto yang diupload oleh seorang user jadi hanya user yang mengupload yang bisa melihat foto yang diupload sedangkan user yang lain tidak bisa melihatnya berikut merupakan tampilan gallery



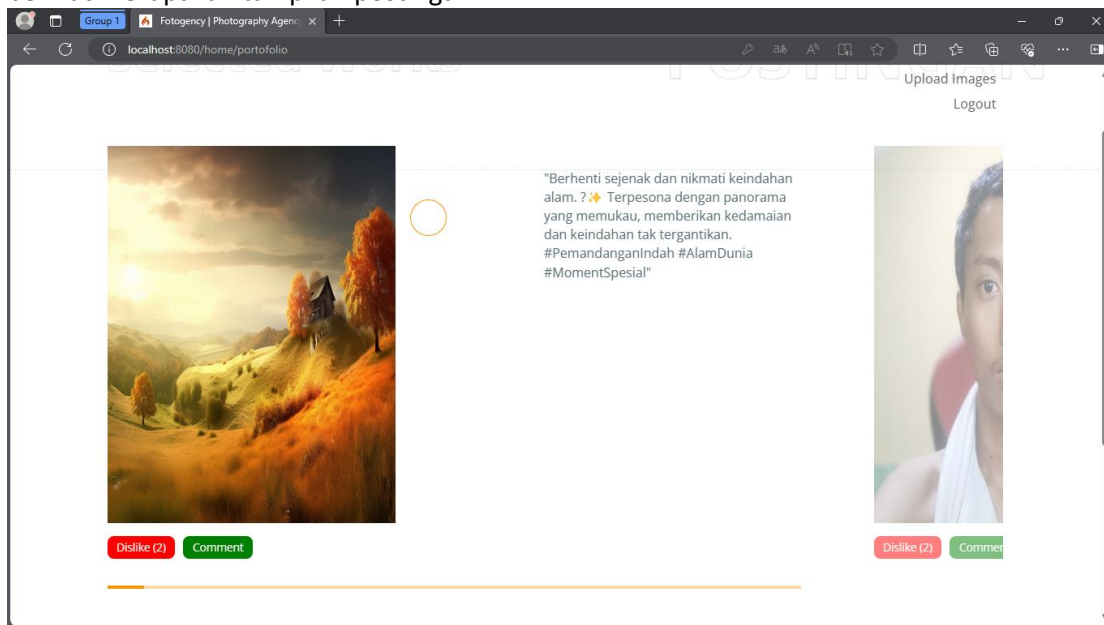
```

public function portofolio()
{
    $model = new M_model();
    $data['projects'] = $model->tampil('photos');
    echo view('header');
    // echo view('menu');
    echo view('portofolio', $data);
    echo view('footer');
}

```

selanjutnya terdapat function portofolio / postingan dengan menggunakan model tampil untuk menampilkan seluruh foto

berikut merupakan tampilan postingan



```

public function comment($postId)
{
    $model = new M_model();
    $postModel = new PostModel();
    $commentModel = new CommentModel();
    $post = $postModel->getPostById($postId); // Menggunakan $postId
    $comments = $commentModel->getCommentsByPost($postId); // Menggunakan $postId

    $data = [
        'post_id' => $postId,
        'photos' => $post,
        'comments' => $comments
    ];

    echo view('header');
    echo view('comment_form', $data);
    echo view('footer');
}

```

selanjutnya terdapat function comment yang tujuannya untuk melakukan comment pada suatu postingan model yang saya gunakan ialah `getPostById` dan `getCommentsByPost`

```
public function getCommentsByPost($postId)
{
    // Menggunakan Query Builder untuk melakukan JOIN dengan tabel user
    return $this->select('comments.*, user.username')
        ->join('user', 'user.id_user = comments.user_id')
        ->where('comments.post_id', $postId)
        ->findAll();
}

public function addComment($data)
{
    return $this->insert($data);
}
```

```
public function getPostById($postId)
{
    return $this->find($postId);
}
```

tujuan dari penggunaan modelnya adalah untuk menampilkan foto yang hanya bisa dilihat saat di click

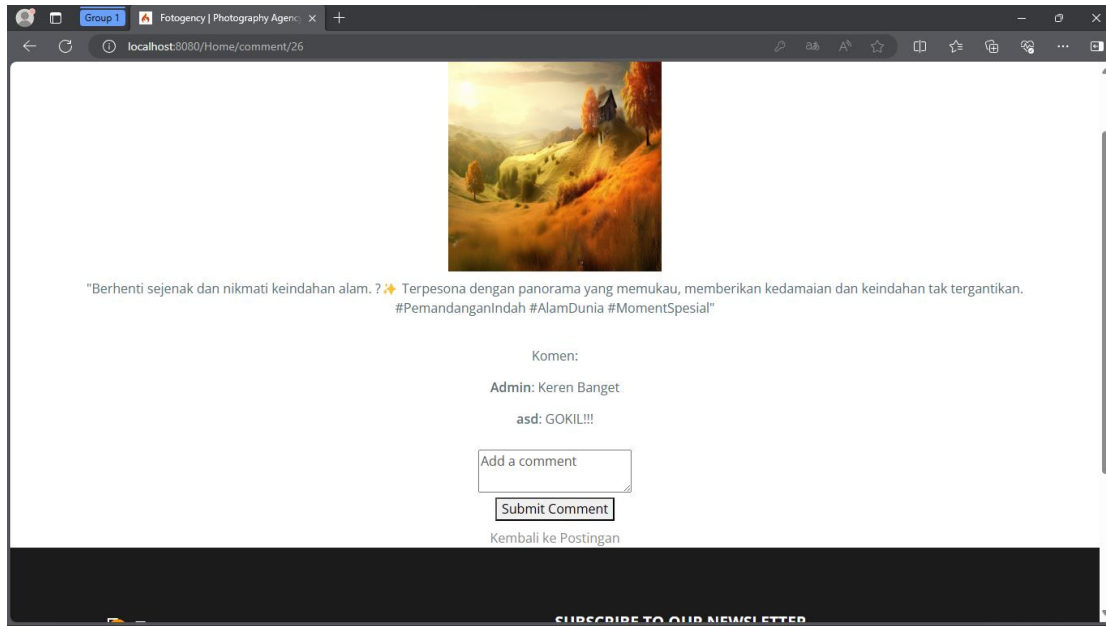
```
public function submitComment()
{
    $commentModel = new CommentModel();
    $userId = session()->get('id_user');
    $postId = $this->request->getPost('post_id');
    $commentText = $this->request->getPost('comment');

    // Simpan komentar ke database menggunakan CommentModel
    $commentModel->addComment(['user_id' => $userId, 'post_id' => $postId, 'comment' => $commentText]);

    // Redirect kembali ke halaman viewAlbum dengan menyertakan id_album
    return redirect()->to(base_url('/home/comment/' . $postId));
}
```

selanjutnya ada function untuk mengeksekusi perintah untuk melakukan comment

berikut merupakan tampilan comment



```
public function upload()
{
    echo view('/header');
    // echo view('menu');
    echo view('/admin/upload');
    // echo view('/footer');
}
```

```

public function aksi_upload()

    $description = $this->request->getPost('description');
    $category = $this->request->getPost('category');
    $user_id = session()->get('id_user');

    $imageFile = $this->request->getFile('image');

    if ($imageFile->isValid() && !$imageFile->hasMoved()) {
        $imageName = $imageFile->getRandomName();
        $imageFile->move('assets/images/', $imageName);

        // Use Intervention Image to resize the image
        $img = Image::make('assets/images/' . $imageName);
        $img->resize(355, 465);
        $img->save('assets/images/' . $imageName);
    } else {
        $imageName = 'default.jpg';
    }

    $data = [
        'user_id' => $user_id, // Assign user_id to the logged-in user's id
        'description' => $description,
        'image' => $imageName,
        'category' => $category,
        'created_at' => date('Y-m-d H:i:s')
    ];

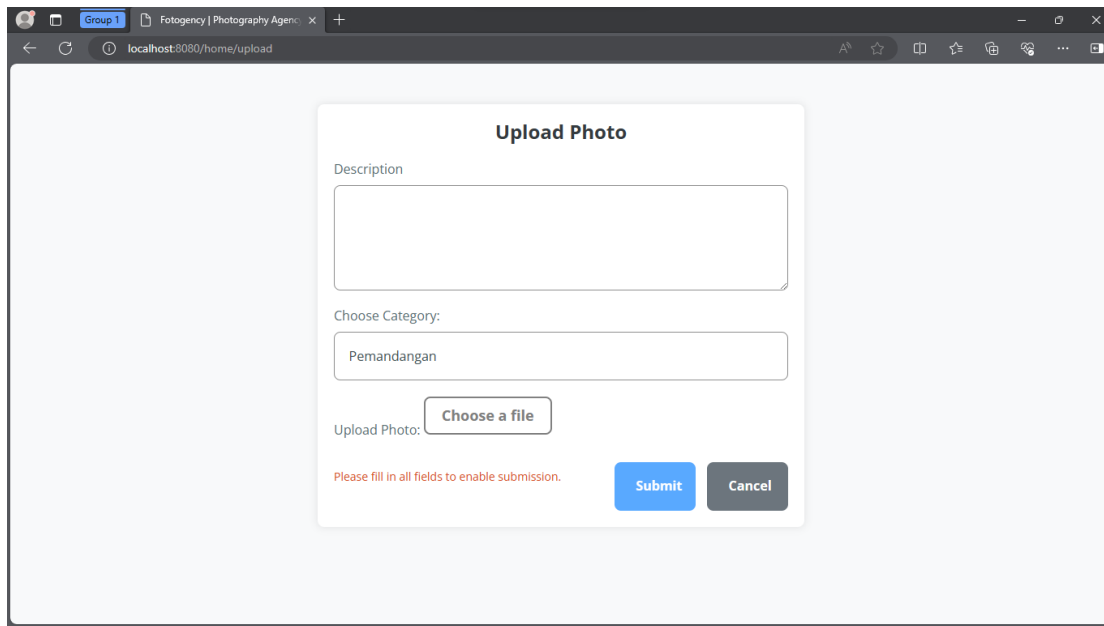
    $model = new M_model();
    $model->simpan('photos', $data);

    return redirect()->to('/home/gallery');

```

kemudian ada function upload dan aksi_upload untuk mengupload suatu foto pada aksi_upload saya menggunakan composer yang berisikan fitur untuk resize sebuah image secara otomatis saat mengupload

berikut merupakan tampilan upload



```
public function register()
{
    $model = new M_model();

    $data['user'] = $model->tampil('user');
    echo view('header');
    // echo view('menuutama');
    echo view('register', $data);
    // echo view('footer');
}
```

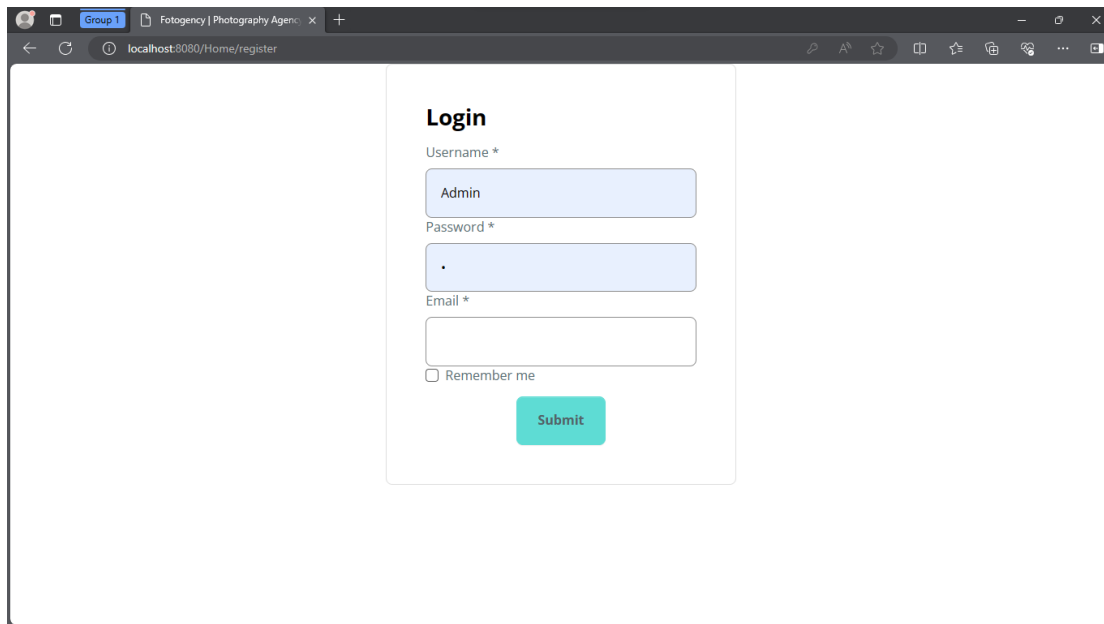
```

public function aksi_register()
{
    $model = new M_model();
    // $on='guru.user = user.id_user';
    $username = $this->request->getPost('username');
    $password = $this->request->getPost('password');
    $email = $this->request->getPost('email');

    $user = array(
        'username' => $username,
        'password' => md5($password),
        'email' => $email,
        'level' => '1',
        'created_at' => date('Y-m-d H:i:s')
    );
    $model = new M_model();
    print_r($user);
    // $model->simpan('user', $user);
    // return redirect()->to('/home');
}
public function aksi_upload()

```

selanjutnya terdapat function register dan aksi register untuk melakukan register bagi calon user
berikut merupakan tampilan register



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/Home/register'. The page content is a login form titled 'Login'. The form contains three input fields: 'Username *' with the value 'Admin', 'Password *' with a single dot, and 'Email *' which is empty. Below the email field is a checkbox labeled 'Remember me' which is unchecked. At the bottom of the form is a green 'Submit' button.

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

kemudian terdapat fitur logout

```
public function hapus($id)
{
    $model = new M_model();

    // Kondisi untuk menghapus data dari tabel 'anggota'
    $where1 = array('id_photos' => $id);

    $model->hapus('photos', $where1);

    return redirect()->to(base_url('/home/gallery'));
}
```

selanjutnya terdapat function hapus untuk menghapus suatu foto