

DOKUMENTASI KODING

SIKULITS



Dipersiapkan oleh:
KELOMPOK A06 – MPPL A

Annas Nuril Iman	(05111740000042)
Najatul Muslim Dinatra	(05111740000079)
Bella Septina Ika Hartanti	(05111740000117)

Jurusan Teknik Informatika - Institut Teknologi Sepuluh Nopember
Kampus ITS Keputih Sukolilo Surabaya 2019

DAFTAR ISI

1	SETTING ENVIRONMENT	4
1.1	Requirement	4
1.2	Clone repo	4
1.3	Membuat database	4
1.4	Mengedit application/config/config.php	4 1.5
	Mengubah base_url	4
2	MODEL	5
2.1	Model Distributor	Error! Bookmark not defined.
2.2	Model Mengganti	7
2.3	Model Menggunakan	7
2.4	Model Motor	8
2.5	Model Pelanggan	9
2.6	Model Service	10
2.7	Model Sparepart	11
2.8	Model Staff	12
2.9	Model Transaksi	14
3	VIEW	17
3.1	LOGIN (login.php)	18
3.2	ADMIN	18
3.2.1	Halaman dashboard.php	18
3.2.2	Halaman distributor.php	Error! Bookmark not defined.
3.2.3	Halaman laporan_keuangan.php	27
3.2.4	Halaman mekanik.php	29
3.2.5	Halaman nota.php	Error! Bookmark not defined.
3.2.6	Halaman pelanggan.php	Error! Bookmark not defined.
3.2.7	Halaman servis.php	
42 3.2.8	Halaman sparepart.php	
47 3.2.9	Halaman transaksi.php	
53 3.2.10	Halaman transaksi_baru.php	
54		
3.3	MEKANIK	56
3.3.1	Halaman dashboard.php	56
3.3.2	Halaman detail_transaksi.php	57

4	CONTROLLER	61
4.1	Dashboard	61
4.2	Laporan Keuangan	62
4.3	Login	63
4.4	Manajemen	63
4.5	Mekanik	70
4.6	Transaksi	72

Aplikasi SIKULITS ini dibangun menggunakan framework Phalcon yang berbasis MVC (Model, View, Controller).

1 SETTING ENVIRONMENT

1.1 Requirement

- OS Windows 8/10
- XAMPP
- Phalcon

1.2 Clone repo

Project SMART-SERVICE dapat di clone di

<https://github.com/bellasih/MPPL>

1.3 Membuat database

Pada repo sudah terdapat script sql untuk membuat database yang diperlukan oleh aplikasi ini. Kita tinggal meng-*import* saja.

1.4 Mengedit application/config/config.php

Ubah hostname, username, password, dan database sesuai konfigurasi database kita

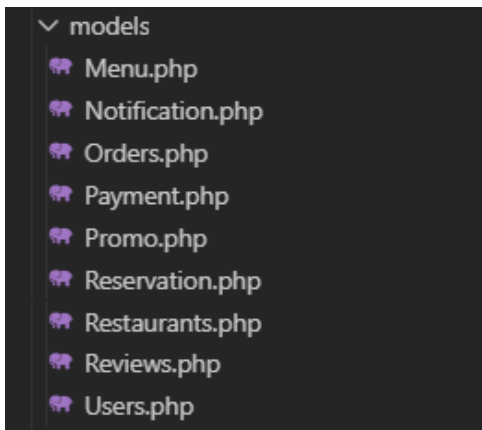
```
app > config > config.php
1  <?php
2
3  use Phalcon\Config;
4
5  return new Config([
6
7      'database' => [
8          'adapter' => 'Phalcon\Db\Adapter\Pdo\Mysql',
9          'host' => '127.0.0.1',
10         'username' => 'root',
11         'password' => '',
12         'dbname' => 'sikulits'
13     ],
```

1.5 Mengubah base_url

Ubah \$config['base_url'] pada application/config/config.php sesuai konfigurasi web server kita

```
14         'url' => [
15             'baseUrl' => 'http://localhost/mppl/'
16         ]
17     ];
```

2 MODEL



Terdapat 9 model, dimana model disini merepresentasikan tabel-tabel pada database.

2.1 Model Menu

```
app > models > Menu.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Menu extends Model
6  {
7      public $id_menu;
8      public $id_restos;
9      public $menu_name;
10     public $price;
11 }
```

2.2 Model Notification

```
app > models > Notification.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Notification extends Model
6  {
7      public $id_notif;
8      public $user;
9      public $msg;
10     public $isread;
11     public $date_notif;
12 }
```

2.2 Model Orders

```
app > models > Orders.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Orders extends Model
6  {
7      public $idorder;
8      public $iduser;
9      public $idresto;
10     public $idmenu;
11     public $amount;
12     public $isvalid;
13 }
```

2.3 Model Payment

```
app > models > Payment.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Payment extends Model
6  {
7      public $idpayment;
8      public $iduser;
9      public $type;
10     public $photo;
11 }
```

2.4 Model Promo

```
app > models > Promo.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Promo extends Model
6  {
7      public $id_promo;
8      public $promo_name;
9      public $category;
10     public $available_in;
11     public $available_start;
12     public $available_until;
13     public $photo_promo;
14     public $last_up;
15     public $description_promo;
16     public $users_id;
17 }
```

2.5 Model Reservation

```
app > models > Reservation.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Reservation extends Model
6  {
7      public $idreserve;
8      public $userid;
9      public $restoid;
10     public $start;
11     public $end;
12     public $amount;
13     public $isvalid;
14 }
```

2.6 Model Restaurant

```
app > models > Restaurants.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Restaurants extends Model
6  {
7      public $id_resto;
8      public $resto_name;
9      public $phone_num;
10     public $location;
11     public $photo;
12     public $omzet;
13     public $user_id;
14     public $date_register;
15     public $isValidate;
16 }
```

2.7 Model Reviews

```
app > models > Reviews.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Reviews extends Model
6  {
7      public $id_review;
8      public $user_id;
9      public $review_contains;
10     public $time_submit;
11     public $resto_id;
12     public $rating;
13 }
```

2.8 Model Users

```
app > models > Users.php
1  <?php
2
3  use Phalcon\Mvc\Model;
4
5  class Users extends Model
6  {
7      public $id;
8      public $username;
9      public $gender;
10     public $email;
11     public $password;
12     public $name;
13     public $phone;
14     public $role;
15 }
```

3 CONTROLLER

```
▼ controllers
  AboutController.php
  EditpromoController.php
  EditrestaurantController.php
  ErrorController.php
  HomeController.php
  OrdersController.php
  PaymentController.php
  ProfileController.php
  PromoController.php
  ReservationController.php
  RestaurantsController.php
  ReviewsController.php
  SessionController.php
  UserController.php
```

Terdapat 14 controller, dimana controller berfungsi untuk proses pengatur antara model dan view.

3.1 About Controller

```
app > controllers > AboutController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5  use Phalcon\Mvc\Dispatcher;
6
7  class AboutController extends Controller
8  {
9      public function indexAction()
10     {
11
12     }
13 }
```


3.2 Editpromo Controller

```
app > controllers > EditpromoController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  // for file uploading
7  use Phalcon\Http\Request\File;
8
9  use App\Forms\PromoForm;
10 use App\Forms>EditpromoForm;
11 use App\Forms\Restaurants;
12
13 class EditpromoController extends Controller
14 {
15     private $error;
16     private $success;
17
18     public function showAction(){
19         if( !$this->session->has('auth') || $this->session->get('auth')['role']!=1)
20             $this->dispatcher->forward(['controller'=>'home', 'action' => 'forbidden']);
21
22         $promo;
23         $i=0;
24         $promos = Promo::find();
25         foreach($promos as $pro){
26             $promo[$i++] = $pro;
27         }
28         $this->view->promos = $promo;
29         $this->view->countpromo = $i;
30         $this->view->error = $this->error;
31         $this->view->success = $this->success;
32     }
33
34     public function deleteAction(){
35         if(!$this->request->isPost()){
36
37             return $this->response->redirect('editpromo');
38         }
39
40         $id_promo = $this->request->getPost('id_promo');
41     }
```

```

        if($id_promo==null){
            $this->error = "Could not find promo's data. Please try again!. '$id_promo'";
        }
        else{
            $promo = Promo::findFirst("id_promo=$id_promo");

            if($promo!=null){
                $fileLoc = BASE_PATH . '///public///' . $promo->photo_promo;
                if($promo->delete()){
                    if(!unlink($fileLoc)) $this->error = 'Photo related to the promo could not be deleted';
                    $this->success = "Promo's data has been deleted. '$id_promo'";
                }
                else $this->error = "An error occurs. Please try again.";
            }
            else{
                $this->error = "Could not find promo's data. Please try again!";
            }
        }
        $this->dispatcher->forward(['action'=>'show']);
    }

    public function editAction(){
        if(!$this->request->hasQuery('id_promo') && !$this->dispatcher ){
            return $this->response->redirect('editpromo');
        }

        $this->view->notif = $this->notif;
        $this->view->error = $this->error;
        $this->view->messages = $this->messages;
        if($this->request->hasQuery('id_promo'))
            $id_promo = $this->request->getQuery('id_promo');

        else if($this->dispatcher)
            $id_promo = $this->dispatcher->getParams()['id_promo'];

        $promo = Promo::findFirst([
            'conditions' => "id_promo= '$id_promo'"
            // 'columns' => "promo_name, category"
        ]);
    }

```

```

83     $this->view->form = $form;
84     $this->view->promo = $promo;
85 }
86
87 public function updateAction(){
88     if(!$this->request->isPost()){
89         $this->response->redirect('edits');
90     }
91
92     $form = new EditpromoForm();
93     if (!$this->session->has('auth'))
94         $this->error = 'You must be logged in.';
95
96     else if(!$form->isValid($this->request->getPost())){
97         foreach ($form->getMessages() as $msg){
98             $this->messages[$msg->getField()] = $msg;
99         }
100     $this->error = 'Could not update promo\'s information';
101 }
102
103 else{
104     $id_promo = $this->request->getPost('id_promo');
105     $promo = Promo::findFirst([
106         'conditions' => "id_promo= '$id_promo'",
107     ]);
108
109     if($promo==null){
110         $this->error = 'Error finding promo\'s data.';
111     }
112
113     else{
114         $postData = $this->request->getPost();
115         $promo->promo_name = $postData['promo_name'];
116         $promo->category = $postData['category'];
117         // $promo->available_in = $postData['available_in'];
118         // $promo->available_start = $postData['available_start'];
119         // $promo->available_until = $postData['available_in'];
120         // $promo->description_promo = $postData['description_promo'];

```

```

        if($promo->update()){
            $this->notif = 'Promo\'s information has been updated';
        }
        else{
            $this->error = 'An error occurred. Please try again!';
        }
    }
    $this->dispatcher->forward(['action'=>'edit', 'params'=>['id_promo'=> $id_promo]]);
}
}

```

3.3 Editrestaurant Controller

```
app > controllers > EditrestaurantController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  // for file uploading
7  use Phalcon\Http\Request\File;
8
9  use App\Forms\RestaurantsForm;
10 use App\Forms\EditrestaurantForm;
11
12 class EditrestaurantController extends Controller
13 {
14     private $error;
15     private $success;
16
17     public function showAction(){
18         if( !$this->session->has('auth') || $this->session->get('auth')['role']!=1)
19             $this->dispatcher->forward(['controller'=>'home', 'action' => 'forbidden']);
20
21         $resto;
22         $i=0;
23         $restos = Restaurants::find();
24         foreach($restos as $res){
25             $resto[$i++] = $res;
26         }
27         $this->view->restos = $resto;
28         $this->view->countresto = $i;
29         $this->view->error = $this->error;
30         $this->view->success = $this->success;
31     }
32
33     public function deleteAction(){
34         if(!$this->request->isPost()){
35             return $this->response->redirect('editresto');
36         }
37         $id_resto = $this->request->getPost('id_resto');
38
39         if($id_resto==null){
40             $this->error = "Could not find restaurant's data. Please try again!";
41         }
42     }
43 }
```

```

41     }
42     else{
43         $restaurant = Restaurants::findFirst("id_resto=$id_resto");
44         if($restaurant!=null){
45             $fileLoc = BASE_PATH . 'public/' . $restaurant->photo;
46             if($restaurant->delete()){
47                 if(!unlink($fileLoc)) $this->error = 'Photo related to the restaurant could not be deleted';
48                 $this->success = "Restaurant's profile has been deleted.";
49             }
50             else $this->error = "An error occurs. Please try again.";
51         }
52         else{
53             $this->error = "Could not find restaurant's data. Please try again!";
54         }
55     }
56     $this->dispatcher->forward(['action'=>'show']);
57 }
58
59 public function editAction(){
60     if(!$this->request->hasQuery('id_resto') && !$this->dispatcher ){
61         return $this->response->redirect('editresto');
62     }
63
64     $this->view->notif = $this->notif;
65     $this->view->error = $this->error;
66     $this->view->messages = $this->messages;
67     if($this->request->hasQuery('id_resto'))
68         $id_resto = $this->request->getQuery('id_resto');
69
70     else if($this->dispatcher)
71         $id_resto = $this->dispatcher->getParams()['id_resto'];
72
73     $restaurant = Restaurants::findFirst([
74         'conditions' => "id_resto= '$id_resto'"
75         // 'columns' => "promo_name, category"
76     ]);
77     //$restaurant = Restaurants::findFirst("id_resto='$id_resto'");
78     $form = new EditrestaurantForm($restaurant);
79     $this->view->form = $form;
80     $this->view->restaurant = $restaurant;

```

```

81     }
82
83     public function updateAction(){
84         if(!$this->request->isPost()){
85             $this->response->redirect('edit');
86         }
87
88         $form = new EditrestaurantForm();
89         if (!$this->session->has('auth'))
90             $this->error = 'You must be logged in.';
91
92         else if (!$form->isValid($this->request->getPost())){
93             foreach ($form->getMessages() as $msg){
94                 $this->messages[$msg->getField()] = $msg;
95             }
96             $this->error = 'Could not update restaurant\'s information';
97         }
98
99         else{
100             $id_resto = $this->request->getPost('id_resto');
101             $resto = Restaurants::findFirst([
102                 'conditions' => "id_resto= '$id_resto'"
103                 // 'columns' => "promo_name, category"
104             ]);
105
106             if($resto==null){
107                 $this->error = 'Error finding restaurant\'s data.';
108             }
109             else{
110                 $postData = $this->request->getPost();
111                 $resto->resto_name = $postData['resto_name'];
112                 $resto->phone_num = $postData['phone_num'];
113                 $resto->location = $postData['location'];
114                 $resto->omzet = $postData['omzet'];
115                 //$resto->date_register = date('Y-m-d');
116
117                 // $menu_res = new Menu();
118                 // $menu_res->id_restos = $postData['id_resto'];
119                 // $menu_res->menu_name = $postData['menu_name'];
120                 // $menu_res->price = $postData['price'];

```

```

116
117         // $menu_res = new Menu();
118         // $menu_res->id_restos = $postData['id_resto'];
119         // $menu_res->menu_name = $postData['menu_name'];
120         // $menu_res->price = $postData['price'];
121
122         if($resto->update()){
123             $this->notif = 'Restaurant\'s information has been updated';
124         }
125         else{
126             $this->error = 'An error occured. Please try again!';
127         }
128
129         // if($menu_res->save()){
130         //     $this->notif = 'Menu\'s information has been added';
131         // }
132         // else{
133         //     $this->error = 'An error occured. Please try again!';
134         // }
135     }
136 }
137 $this->dispatcher->forward(['action'=>'edit', 'params'=>['id_resto'=> $id_resto]]);
138 }
139 }

```

3.4 Error Controller

```

app > controllers > ErrorController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  class ErrorController extends Controller
7  {
8      public function show500Action()
9      {
10     }
11
12     public function show404Action()
13     {
14     }
15
16     public function forbiddenAction()
17     {
18     }
19
20 }

```

3.5 Home Controller

```
app > controllers > HomeController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  class HomeController extends Controller
7  {
8      public function indexAction()
9      {
10
11      }
12
13      public function show404Action()
14      {
15      }
16
17      public function forbiddenAction()
18      {
19      }
20
21  }
```

3.6 Orders Controller

```
app > controllers > OrdersController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  use Phalcon\Http\Request\File;
7  use App\Forms\OrdersForm;
8
9  class OrdersController extends Controller
10 {
11     public $notif;
12     public $error;
13
14     public function initialize(){
15         $this->messages = [
16             'resto_name' => '',
17             'menu_name' => ''
18         ];
19         $this->notif = "";
20         $this->error = "";
21     }
22
23
24     public function createAction()
25     {
26         $this->view->notif = $this->notif;
27         $this->view->error = $this->error;
28         $this->view->messages = $this->messages;
29         $this->view->form = new OrdersForm();
30     }
31
32     public function storeAction()
33     {
34         if(!$this->request->isPost()){
35             $this->response->redirect('order');
36         }
37
38         $form = new OrdersForm();
39         if (!$this->session->has('auth')) {
40             $this->error = 'You must be logged in.';
41             $this->dispatcher->forward(['controller'=>'home', 'action' => 'forbidden']);
42         }
43     }
44 }
```



```

43
44     if(!$form->isValid($this->request->getPost())){
45         foreach ($form->getMessages() as $msg){
46             $this->messages[$msg->getField()] = $msg;
47         }
48     }
49
50     if($this->message['resto_name']==null && $this->message['menu_name']==null){
51         $order = new Orders();
52         $resto_name = $this->request->getPost('resto_name');
53         $menu_name = $this->request->getPost('menu_name');
54
55         $resto_data = Restaurants::findFirst("resto_name = '$resto_name'");
56         $menu_data = Menu::findFirst("menu_name = '$menu_name' AND id_restos = '$resto_data->id_resto'");
57
58         $order->iduser = $this->session->get('auth')['id'];
59         $order->idresto = $resto_data->id_resto;
60         $order->idmenu = $menu_data->id_menu;
61         $order->amount = $menu_data->price;
62         $order->isvalid = 0;
63
64         if($order->save()){
65             $this->notif = 'Order successfully submitted';
66         }
67         else{
68             $this->error = "An error occurred. Please try again.";
69         }
70     }
71
72     else $this->error = 'Could not submit the order. Please fill the form!';
73
74     $this->dispatcher->forward(['action'=>'create']);
75 }

```

```

77     public function validateAction()
78     {
79         $order;
80         $i=0;
81         $orders = Orders::find("isvalid = 0");
82         foreach($orders as $o){
83             $order[$i++] = $o;
84         }
85         $this->view->orders = $order;
86         $this->view->count = $i;
87         $this->view->error = $this->error;
88         $this->view->success = $this->success;
89     }
90 }

```

3.7 Payment Controller

```
app > controllers > PaymentController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  // for file uploading
7  use Phalcon\Http\Request\File;
8  use App\Forms\PaymentForm;
9
10
11 class PaymentController extends Controller
12 {
13     public $notif;
14     public $error;
15     public $messages;
16
17     public function initialize(){
18         $this->messages = [
19             'type' => '',
20             'payment_photo' => ''
21         ];
22         $this->notif = "";
23         $this->error = "";
24     }
25     public function createAction()
26     {
27         $this->view->notif = $this->notif;
28         $this->view->error = $this->error;
29         $this->view->messages = $this->messages;
30         $this->view->form = new PaymentForm();
31     }
32
33     public function storeAction()
34     {
35         if(!$this->request->isPost()){
36             $this->response->redirect();
37         }
38     }
39 }
```

```

33 public function storeAction()
34 {
35     if(!$this->request->isPost()){
36         $this->response->redirect();
37     }
38
39     $form = new RestaurantsForm();
40     if (!$this->session->has('auth')) {
41         $this->error = 'You must be logged in.';
42         $this->dispatcher->forward(['action' => 'payment']);
43     }
44
45     if(!$form->isValid($this->request->getPost())){
46         foreach ($form->getMessages() as $msg){
47             if($msg->getField()!='payment_photo')
48                 $this->messages[$msg->getField()] = $msg;
49         }
50     }
51
52     if( $this->request->hasFiles('payment_photo') && $this->messages['type']==null)
53     {
54         $pay = new Payment();
55
56         if ($this->request->hasFiles() == true) {
57             $pay->iduser = $this->session->get('auth')['id'];
58             $pay->type = $this->request->getPost('type');
59             $pay->photo = 'temp.jpg';
60
61             if($pay->save()){
62                 $this->notif = 'Paymment successfully submitted';
63                 foreach ($this->request->getUploadedFiles() as $file) {
64                     $file_toDB = "img\\payment\\" . $pay->idpayment . '.' . $file->getExtension();
65                     $target_file = BASE_PATH . '\\public\\' . $file_toDB;
66                     $file->moveTo($target_file);
67                     $payment->photo = $file_toDB;
68                     $payment->update();
69                 }
70             }

```

```

        else $this->error = 'An error occured. Please try again.';
    }
    else $this->error = 'Could not report the restaurant. Please upload a photo of the restaurant.';
}
else $this->error = 'Couldn\'t attach the photo';

$this->dispatcher->forward(['action'=>'create']);
}

public function listAction()
{
    $resto;
    $menu;
    $menus;
    $i=0;
    $restos = Restaurants::find();
    foreach($restos as $r){
        $resto[$i++] = $r;
        $menus = Menu::find("id_restos = $r->id_resto");
        $menu[$i] = $menus;
    }
    $this->view->restos = $resto;
    $this->view->menus = $menu;
    $this->view->count = $i;
    $this->view->error = $this->error;
    $this->view->success = $this->success;
}
}

```

3.8 Profile Controller

```
app > controllers > ProfileController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  use App\Forms\SignupForm;
7  use App\Forms\UploadForm;
8  use Phalcon\Forms\Form;
9  use Phalcon\Forms\Element\File;
10 use Phalcon\Forms\Element\Submit;
11
12 use Phalcon\Validation;
13 use Phalcon\Validation\Validator\File as FileValidator;
14
15 class ProfileController extends Controller
16 {
17     private $messages;
18
19     public function initialize(){
20         $this->messages = [
21             'error'      => '',
22             'warning'    => '',
23             'success'    => '',
24             'photo_user' => '',
25             'name'       => '',
26             'username'   => '',
27             'email'      => '',
28             'phone'      => '',
29             'password'   => '',
30             'confirmp'   => ''
31         ];
32     }
33
34
35
36     private function createProfile($us){
37         // authorization
38         if( !$this->session->has('auth'))
39             $this->dispatcher->forward(['controller'=>'home', 'action' => 'forbidden']);
40
41     }
```

```

40
41
42     $user = $us;
43     $user->password='';
44     $form = new SignupForm($user);
45
46     $form->add(
47         new Submit(
48             'Save',
49             [
50                 'class' => 'btn btn-info',
51                 'type' => 'button'
52             ]
53         )
54     );
55
56     return $form;
57 }
58
59
60
61 public function showAction()
62 {
63     $username = $this->session->get('auth')['username'];
64     $user = Users::findFirst("username='$username'");
65     $this->view->user = $user;
66     $this->view->profile = $this->createProfile($user);
67     $this->view->upload = new UploadForm();
68     $this->view->messages = $this->messages;
69
70     // notification
71     $query = $this->modelsManager->createQuery('SELECT * FROM Notification WHERE user = :username: and isread=0 c
72     $user_notif = $query->execute(
73         [
74             'username' => $user->username,
75         ]
76     );
77
78     $this->view->user_notif = $user_notif;
79
80 }

```

```

public function save_nameAction(){
    if(!$this->request->isPost())
    {
        return $this->response->redirect('profile');
    }

    if($this->request->getPost('name')=='){
        $this->messages['warning'] = 'No changes was made.';
        return $this->dispatcher->forward(['action'=>'show']);
    }

    else{
        $username = $this->session->get('auth')['username'];
        $user = Users::findFirst("username='$username'");

        $user->name = $this->request->getPost('name');
        if($user->update()){
            $this->messages['success'] = 'Changes has been saved';
        }
        else{
            $this->messages['error'] = 'An error occurs. Please try again.';
        }
        return $this->dispatcher->forward(['action'=>'show']);
    }
}
}

```

```

109
110 ✓ public function save_accountAction(){
111     if(!$this->request->isPost())
112     {
113         return $this->response->redirect('profile');
114     }
115     $username = $this->session->get('auth')['username'];
116     $user = Users::findFirst("username='$username'");
117     $form = new SignupForm($user);
118
119 ✓     if(!$form->isValid($this->request->getPost())){
120 ✓         foreach($form->getMessages() as $msg){
121 ✓             if(in_array ($msg->getField(), ['username', 'email', 'phone']))
122                 $this->messages[$msg->getField()] = $msg;
123         }
124     }
125
126 ✓     if($this->messages['username']=='' & $this->messages['email']=='' & $this->messages['phone']==''){
127
128         $tobeUsername = $this->request->getPost('username');
129
130 ✓         if($tobeUsername!=$username & Users::findFirst("username='$tobeUsername'")!=null){
131             $this->messages['error'] = 'Could not save changes';
132         }
133 ✓         else{
134             $user->username = $tobeUsername;
135             $user->email     = $this->request->getPost('email');
136             $user->phone     = $this->request->getPost('phone');
137
138 ✓             if($user->update()){
139                 $this->messages['success'] = 'Changes has been saved';
140             }
141 ✓             else{
142                 $this->messages['error'] = 'An error occurs. Please Try again later.';
143             }
144         }
145     }
146 ✓     else
147         $this->messages['error'] = 'Could not save changes.';
148

```

```

146     else
147         $this->messages['error'] = 'Could not save changes.';
148
149     $this->dispatcher->forward(['action' => 'show']);
150 }
151
152 public function save_passwordAction(){
153     if(!$this->request->isPost())
154     {
155         return $this->response->redirect('profile');
156     }
157
158     $username = $this->session->get('auth')['username'];
159     $user = Users::findFirst("username='$username'");
160     $form = new SignupForm($user);
161
162     if(!$form->isValid($this->request->getPost())){
163         foreach($form->getMessages() as $msg){
164             if(in_array ($msg->getField(), ['password', 'confirmp']))
165                 $this->messages[$msg->getField()] = $msg;
166         }
167     }
168

```

```

195 public function save_photoAction(){
196     if(!$this->request->isPost())
197     {
198         return $this->response->redirect('profile');
199     }
200
201     $username = $this->session->get('auth')['username'];
202     $user = Users::findFirst("username='$username'");
203     $form = new UploadForm();
204
205     if(!$form->isValid($_FILES)){
206         foreach($form->getMessages() as $msg){
207             $this->messages['photo_user'] = $msg;
208         }
209     }
210
211     if($this->messages['photo_user']==='') {
212         $file = null;
213
214         $file = file_get_contents($_FILES['photo_user']['tmp_name']);
215         $enc_file = base64_encode($file);
216         $user->photo = $enc_file;
217
218         if($user->update()){
219             $this->messages['success'] = 'Changes has been saved';
220         }
221         else{
222             $this->messages['error'] = 'An error occurs. Please Try again later.';
223         }
224     }
225     else{
226         $this->messages['error'] = 'Could not save changes.';
227     }
228     $this->dispatcher->forward(['action' => 'show']);
229 }
230
231

```

```

232 public function seeAction(){
233     if(!$this->request->hasQuery('id_notif') || !$this->session->has('auth')){
234         $this->dispatcher->forward(['controller'=>'home', 'action'=>'forbidden']);
235     }
236
237     $id_notif = $this->request->getQuery('id_notif');
238     $user_notif = Notification::findFirst("id_notif='$id_notif'");
239
240     // if notif doesn't belong to user, forbidden
241     if($this->session->get('auth')['username']!=$user_notif->user)
242         $this->dispatcher->forward(['controller'=>'home', 'action'=>'forbidden']);
243
244     // $cat = Cats::findFirst("id_cat = '$user_notif->cat'");
245
246     // $this->view->cat = $cat;
247 }
248

```


3.9 Promo Controller

```
app > controllers > PromoController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  // for file uploading
7  use Phalcon\Http\Request\File;
8  use App\Forms\PromoForm;
9
10
11 class PromoController extends Controller
12 {
13     public $notif;
14     public $error;
15     public $messages;
16
17     public function initialize()
18     {
19         $this->messages = [
20             'promo_name' => '',
21             'category' => '',
22             'available_in' => '',
23             'available_start' => '',
24             'available_until' => '',
25             'description_promo' => '',
26             'photo_promo' => ''
27         ];
28         $this->notif = "";
29         $this->error = "";
30     }
31     public function createAction()
32     {
33         //authorization
34         if($this->session->get('auth')['role']!=1)
35             $this->dispatcher->forward(['action' => 'list']);
36
37         $this->view->notif = $this->notif;
38         $this->view->error = $this->error;
39         $this->view->messages = $this->messages;
40         $this->view->form = new PromoForm();
41     }
42 }
```

```

42 public function storeAction()
43 {
44     if(!$this->request->isPost()){
45         $this->response->redirect();
46     }
47
48     $form = new PromoForm();
49     if (!$this->session->has('auth')) {
50         $this->error = 'You must be logged in.';
51         $this->dispatcher->forward(['action' => 'promo']);
52     }
53
54     if(!$form->isValid($this->request->getPost())){
55         foreach ($form->getMessages() as $msg){
56             if($msg->getField()!='photo_promo')
57                 $this->messages[$msg->getField()] = $msg;
58         }
59     }
60
61     if( $this->request->hasFiles('photo_promo') )
62     {
63         $promos = new Promo();
64         if ($this->request->hasFiles() == true) {
65             $promos->promo_name = $this->request->getPost('promo_name');
66             $promos->category = $this->request->getPost('category');
67             $promos->available_in = $this->request->getPost('available_in');
68             $promos->available_start = $this->request->getPost('available_start');
69             $promos->available_until = $this->request->getPost('available_until');
70             $promos->last_up = date('Y-m-d');
71             $promos->users_id = $this->session->get('auth')['id'];
72             $promos->photo_promo = 'temp_promo.jpg';
73             $promos->description_promo = $this->request->getPost('description_promo');
74

```

```

75         if($promos->save()){
76             $this->notif = 'Promo successfully created';
77             foreach ($this->request->getUploadedFiles() as $file) {
78                 $file_toDB = "img\\img_promo\\" . $promos->id_promo . '.' . $file->getExtension();
79                 $target_file = BASE_PATH . '\\public\\' . $file_toDB;
80                 $file->moveTo($target_file);
81                 $promos->photo_promo = $file_toDB;
82                 $promos->update();
83             }
84         }
85         else $this->error = 'An error occured. Please try again!';
86     }
87     else $this->error = 'Could not add the promo. Please upload a photo of the promo.';
88 }
89 $this->dispatcher->forward(['action'=>'create']);
90 }
91
92 public function listAction()
93 {
94     $dis;
95     $i=0;
96     $promos = Promo::find();
97     foreach($promos as $p){
98         $dis[$i++] = $p;
99     }
100     $this->view->promos = $dis;
101     $this->view->count = $i;
102     $this->view->error = $this->error;
103     $this->view->success = $this->success;
104 }
105 }

```

3.10 Reservation Controller

```

app > controllers > ReservationController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  use Phalcon\Http\Request\File;
7  use App\Forms\ReservationForm;
8
9  class ReservationController extends Controller
10 {
11     public $notif;
12     public $error;
13
14     public function initialize(){
15         $this->messages = [
16             'resto_name' =>'',
17             'start' =>'',
18             'end' =>'',
19             'amount' =>''
20         ];
21         $this->notif = '';
22         $this->error = '';
23     }
24
25     public function createAction()
26     {
27         $this->view->notif = $this->notif;
28         $this->view->error = $this->error;
29         $this->view->messages = $this->messages;
30         $this->view->form = new ReservationForm();
31     }
32
33     public function storeAction()
34     {
35         if(!$this->request->isPost()){
36             $this->response->redirect();
37         }
38     }

```

```

39     $form = new ReservationForm();
40     if (!$this->session->has('auth')) {
41         $this->error = 'You must be logged in.';
42         $this->dispatcher->forward(['action' => 'reservation']);
43     }
44
45     if (!$form->isValid($this->request->getPost())){
46         foreach ($form->getMessages() as $msg){
47             $this->messages[$msg->getField()] = $msg;
48         }
49     }
50
51     if($this->message['resto_name']==null && $this->message['start']==null && $this->message['end']==null &
52         $reserve = new Reservation();
53         $resto_name = $this->request->getPost('resto_name');
54
55         $resto_data = Restaurants::findFirst("resto_name = '$resto_name'");
56
57         $reserve->userid = $this->session->get('auth')['id'];
58         $reserve->restoid = $resto_data->id_resto;
59         $reserve->start = $this->request->getPost('start');
60         $reserve->end = $this->request->getPost('end');
61         $reserve->amount = $this->request->getPost('amount');
62         $reserve->isvalid = 0;
63
64         if($reserve->save()){
65             $this->notif = 'Reservation successfully submitted';
66         }
67         else $this->error = 'An error occured. Please try again.';
68     }
69     else $this->error = 'Could not submit the order. Please fill the form!';
70
71     $this->dispatcher->forward(['action'=>'create']);
72 }
73

```

```

74     public function validateAction()
75     {
76         $reserve;
77         $i=0;
78         $reserves = Reservation::find("isvalid = 0");
79         foreach($reserves as $res){
80             $reserve[$i++] = $res;
81         }
82         $this->view->reserves = $reserve;
83         $this->view->count = $i;
84         $this->view->error = $this->error;
85         $this->view->success = $this->success;
86     }
87 }

```

3.11 Restaurant Controller

```
app > controllers > RestaurantsController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5
6  // for file uploading
7  use Phalcon\Http\Request\File;
8  use App\Forms\RestaurantsForm;
9
10
11 class RestaurantsController extends Controller
12 {
13     public $notif;
14     public $error;
15     public $messages;
16
17     public function initialize(){
18         $this->messages = [
19             'resto_name' => '',
20             'phone_num' => '',
21             'resto_photo' => '',
22             'location' => '',
23             'omzet' => ''
24         ];
25         $this->notif = "";
26         $this->error = "";
27     }
28     public function createAction()
29     {
30         $this->view->notif = $this->notif;
31         $this->view->error = $this->error;
32         $this->view->messages = $this->messages;
33         $this->view->form = new RestaurantsForm();
34     }
```

```

28 public function createAction()
29 {
30     $this->view->notif = $this->notif;
31     $this->view->error = $this->error;
32     $this->view->messages = $this->messages;
33     $this->view->form = new RestaurantsForm();
34 }
35
36 public function storeAction()
37 {
38     if(!$this->request->isPost()){
39         $this->response->redirect();
40     }
41
42     $form = new RestaurantsForm();
43     if (!$this->session->has('auth')) {
44         $this->error = 'You must be logged in.';
45         $this->dispatcher->forward(['action' => 'restaurants']);
46     }
47
48     if(!$form->isValid($this->request->getPost())){
49         foreach ($form->getMessages() as $msg){
50             if($msg->getField()!='resto_photo')
51                 $this->messages[$msg->getField()] = $msg;
52         }
53     }
54
55     if( $this->request->hasFiles('resto_photo') && $this->messages['resto_name']==null && $this->messages['phone_
56         $this->messages['location']==null && $this->messages['omzet']==null)
57     {
58         $resto = new Restaurants();
59
60         if ($this->request->hasFiles() == true) {
61             $resto->resto_name = $this->request->getPost('resto_name');
62             $resto->phone_num = $this->request->getPost('phone_num');
63             $resto->location = $this->request->getPost('location');
64             $resto->omzet = $this->request->getPost('omzet');
65             $resto->user_id = $this->session->get('auth')['id'];

```

3.12 Review Controller

```
app > controllers > ReviewsController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5  use Phalcon\Mvc\Dispatcher;
6
7  use App\Forms\ReviewsForm;
8
9  class ReviewsController extends Controller
10 {
11     public $notif;
12     public $error;
13     public $messages;
14
15     public function initialize(){
16         $this->messages = [
17             'alias' => '',
18             'review_contains' => '',
19             'rating' => ''
20         ];
21         $this->notif = "";
22         $this->error = "";
23     }
24
25     public function createAction()
26     {
27         $param = $this->request->getQuery('id_resto', 'int');
28         $review;
29         $reviews = Reviews::find("resto_id = '$param'");
30         $i=0;
31         foreach($reviews as $rev){
32             $review[$i++] = $rev;
33         }
34         $this->view->reviews = $review;
35         $this->view->count = $i;
36         $this->view->messages = $this->messages;
37         $this->view->error = $this->error;
38         $this->view->notif = $this->notif;
39         $this->view->form = new ReviewsForm();
40     }
```

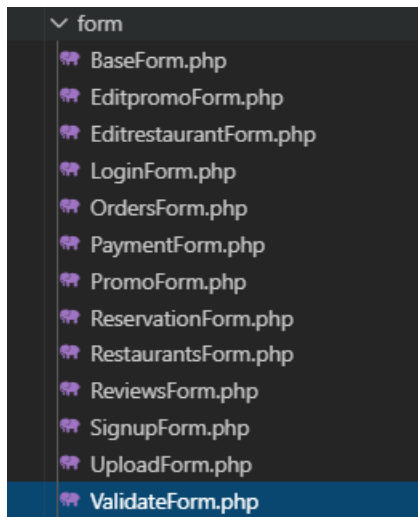
3.13 Session Controller

```
app > controllers > SessionController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5  use Phalcon\Mvc\Dispatcher;
6
7  use App\Forms\LoginForm;
8
9
10 class SessionController extends Controller
11 {
12     private $message = "";
13
14     public function createAction()
15     {
16         if($this->session->has('auth'))
17             $this->response->redirect('profile');
18
19         $form = new LoginForm();
20
21         // Check if the cookie has previously set
22         if ($this->cookies->has('remember-username') && $this->cookies->has('remember-password')) {
23             // Get the cookie's values
24             $username = $this->cookies->get('remember-username')->getValue();
25             $password = $this->cookies->get('remember-password')->getValue();
26
27             // set value of form
28             $attr = $form->get('username')->getAttributes();
29             $attr['value'] = $username;
30             $form->get('username')->setAttributes($attr);
31
32             $attr = $form->get('password')->getAttributes();
33             $attr['value'] = $password;
34             $form->get('password')->setAttributes($attr);
35         }
36         //////////////////////////////////////
37
38         $this->view->message=$this->message;
39         $this->view->form = $form;
40     }
}
```


3.14 User Controller

```
app > controllers > 📁 UserController.php
1  <?php
2
3  use Phalcon\Mvc\Controller;
4  use Phalcon\Http\Response;
5  use App\Forms\SignupForm;
6
7
8  class UserController extends Controller
9  {
10     public $notif;
11     public $error;
12
13     public function initialize(){
14         $this->message = [
15             'username' => '',
16             'name' => '',
17             'gender' => '',
18             'email' => '',
19             'password' => '',
20             'confirmp' => '',
21             'phone' => ''
22         ];
23     }
24
25     private function preparePhoto(){
26         $imageloc = APP_PATH . '\\img\\default_profile.png';
27         $image = file_get_contents($imageloc);
28         $image = base64_encode($image);
29         return $image;
30     }
31
32     public function createAction()
33     {
34         if($this->session->has('auth'))
35             $this->dispatcher->forward(['controller'=>'home', 'action' => 'forbidden']);
36
37         $this->view->form = new SignupForm();
38         $this->view->message = $this->message;
39         $this->view->notif = $this->notif;
40         $this->view->error= $this->error;
41     }
42 }
```

4 FORM



Terdapat 13 controller, dimana controller berfungsi untuk proses pengatur antara model dan view.

4.1 Base Form

```
app > form > BaseForm.php
1  <?php
2
3  Namespace App\Forms;
4
5  use Phalcon\Forms\Form;
6  use Phalcon\Tag;
7
8  class BaseForm extends Form{
9      public function rendering($nameForm){
10         return $this->render($nameForm);
11     }
12
13     public function startForm(){
14         return Tag::form($this->getUserOptions());
15     }
16
17     public function endForm(){
18         return Tag::endForm();
19     }
20 }
21
22
```

4.2 Edit promo Form

```
app > form > EditpromoForm.php
1  <?php
2  namespace App\Forms;
3
4  use App\Forms\BaseForm;
5  use App\Forms\Restaurants;
6
7  use Phalcon\Forms\Form;
8  use Phalcon\Forms\Element\Text;
9  use Phalcon\Forms\Element\Hidden;
10 use Phalcon\Forms\Element\Number;
11 use Phalcon\Forms\Element\TextArea;
12 use Phalcon\Forms\Element\File;
13 use Phalcon\Forms\Element\Submit;
14 use Phalcon\Forms\Element\Select;
15
16 use Phalcon\Tag;
17
18 // validation
19 use Phalcon\Validation;
20 use Phalcon\Validation\Validator\PresenceOf;
21 use Phalcon\Validation\Validator\Alpha;
22 use Phalcon\Validation\Validator\StringLength;
23 use Phalcon\Validation\Validator\InclusionIn;
24
25
26 class EditpromoForm extends BaseForm {
27     public function initialize(){
28         // ----- id resto -----
29         $id = new Hidden('id_promo');
30
31         // ----- restaurant's name -----
32
33         $name_resto = new Text( 'promo_name',
34             [
35                 'placeholder' => 'Enter Restaurant\'s Name',
36                 'class' => 'form-control'
37             ]);
38         $name_resto->setLabel("Enter restaurant's name");
39         $name_resto->addValidator(new PresenceOf([
40             'message'=>'Restaurant\'s name required'
41         ]));
42     }
43 }
```

4.3 Edit restaurant Form

```
app > form > EditrestaurantForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\Text;
8  use Phalcon\Forms\Element\Hidden;
9  use Phalcon\Forms\Element\Number;
10 use Phalcon\Forms\Element\TextArea;
11 use Phalcon\Forms\Element\File;
12 use Phalcon\Forms\Element\Submit;
13
14 use Phalcon\Tag;
15
16 // validation
17 use Phalcon\Validation;
18 use Phalcon\Validation\Validator\PresenceOf;
19 use Phalcon\Validation\Validator\Alpha;
20 use Phalcon\Validation\Validator\StringLength;
21 use Phalcon\Validation\Validator\InclusionIn;
22
23
24 class EditrestaurantForm extends BaseForm {
25     public function initialize(){
26         // ----- id resto -----
27         $id = new Hidden('id_resto');
28
29         // ----- restaurant's name -----
30
31         $name_resto = new Text( 'resto_name',
32         [
33             'placeholder' => 'Enter Restaurant\'s Name',
34             'class' => 'form-control'
35         ]);
36         $name_resto->setLabel("Enter restaurant's name");
37         $name_resto->addValidator(new PresenceOf([
38             'message'=>'Restaurant\'s name required'
39         ]));
40     }
```

4.4 Login Form

```
app > form > LoginForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\Text;
8  use Phalcon\Forms\Element>Password;
9  use Phalcon\Forms\Element\Check;
10 use Phalcon\Forms\Element\Submit;
11 use Phalcon\Forms\Element\Hidden;
12
13 use Phalcon\Validation;
14 use Phalcon\Validation\Validator\Alnum;
15
16 use Phalcon\Tag;
17
18 class LoginForm extends BaseForm {
19     public function initialize(){
20         $id_user = new Hidden('id',
21             [
22                 'value' => 0
23             ]
24         );
25
26         $username = new Text ('username',
27             [
28                 "placeholder" => "Enter Username",
29                 "class" => "form-control"
30             ]
31         );
32
33         $username->addValidator(new Alnum(
34             [
35                 'message' => 'Username must consist of alphanumeric only'
36             ]
37         ));
38
39         $password = new Password ('password',
40             [
41                 "placeholder" => "Enter password",
42                 "class" => "form-control"
```


4.5 Order Form

```
app > form > OrdersForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\Text;
8  use Phalcon\Forms\Element\TextArea;
9  use Phalcon\Forms\Element\File;
10 use Phalcon\Forms\Element\Submit;
11
12 use Phalcon\Tag;
13
14 // validation
15 use Phalcon\Validation;
16 use Phalcon\Validation\Validator\PresenceOf;
17 use Phalcon\Validation\Validator\Alpha;
18 use Phalcon\Validation\Validator\StringLength;
19 use Phalcon\Validation\Validator\InclusionIn;
20
21
22 class OrdersForm extends BaseForm {
23     public function initialize(){
24
25         // ----- restaurant's name -----
26
27         $name_resto = new Text( 'resto_name',
28             [
29                 'placeholder' => 'Enter Restaurant\'s Name',
30                 'class' => 'form-control'
31             ]
32         );
33         $name_resto->setLabel("Enter restaurant's name");
34         $name_resto->addValidator(new PresenceOf([
35             'message'=>'Restaurant\'s name required'
36         ]));
37
38         // ----- restaurant's phone number -----
39
40         $menu_name = new Text('menu_name',
41             [
42                 'placeholder' => 'Enter Menu',
```

4.6 Payment Form

```
app > form > PaymentForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\Text;
8  use Phalcon\Forms\Element\TextArea;
9  use Phalcon\Forms\Element\File;
10 use Phalcon\Forms\Element\Submit;
11
12 use Phalcon\Tag;
13
14 // validation
15 use Phalcon\Validation;
16 use Phalcon\Validation\Validator\PresenceOf;
17 use Phalcon\Validation\Validator\Alpha;
18 use Phalcon\Validation\Validator\StringLength;
19 use Phalcon\Validation\Validator\InclusionIn;
20
21
22 class PaymentForm extends BaseForm {
23     public function initialize(){
24
25         // ----- type of payment -----
26
27         $type = new Text( 'type',
28             [
29                 'placeholder' => 'Order or Reservation',
30                 'class' => 'form-control'
31             ]
32         );
33         $type->setLabel("Payment for");
34         $type->addValidator(new PresenceOf([
35             'message'=>'Payment for required'
36         ]));
37
38         //----- payment evidence -----
39
40         $payment_photo = new File('payment_photo',
41             [
42                 'placeholder' => 'Browse Picture',
```

4.7 Promo Form

```
app > form >  PromoForm.php
4   use App\Forms\BaseForm;
5
6   use Phalcon\Forms\Form;
7   use Phalcon\Forms\Element\Text;
8   use Phalcon\Forms\Element\TextArea;
9   use Phalcon\Forms\Element\File;
10  use Phalcon\Forms\Element\Submit;
11  use Phalcon\Forms\Element\Date;
12
13  use Phalcon\Tag;
14
15  // validation
16  use Phalcon\Validation;
17  use Phalcon\Validation\Validator\PresenceOf;
18  use Phalcon\Validation\Validator\Alpha;
19  use Phalcon\Validation\Validator\StringLength;
20  use Phalcon\Validation\Validator\InclusionIn;
21
22
23  class PromoForm extends BaseForm {
24      public function initialize(){
25
26          // ----- promo's code -----
27
28          $promo_code = new Text('promo_name',
29              [
30                  'placeholder' => 'Enter promo\'s code',
31                  'class' => 'form-control'
32              ]);
33          $promo_code->setLabel('Enter promo\'s code');
34          $promo_code->addValidator(new PresenceOf([
35              'message'=>'Promo\'s code required'
36          ]));
37
38          // ----- promo's category -----
39          $promo_cat = new Text('category',
40              [
41                  'placeholder' => 'Enter promo\'s category',
42                  'class' => 'form-control'
43              ]
44          );
```


4.8 Reservation Form

```
app > form > ReservationForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\Text;
8  use Phalcon\Forms\Element\TextArea;
9  use Phalcon\Forms\Element\File;
10 use Phalcon\Forms\Element\Submit;
11 use Phalcon\Forms\Element\Date;
12
13 use Phalcon\Tag;
14
15 // validation
16 use Phalcon\Validation;
17 use Phalcon\Validation\Validator\PresenceOf;
18 use Phalcon\Validation\Validator\Alpha;
19 use Phalcon\Validation\Validator\StringLength;
20 use Phalcon\Validation\Validator\InclusionIn;
21
22
23 class ReservationForm extends BaseForm {
24     public function initialize(){
25
26         // ----- restaurant's name -----
27
28         $name_resto = new Text( 'resto_name',
29         [
30             'placeholder' => 'Enter Restaurant\'s Name',
31             'class' => 'form-control'
32         ]);
33         $name_resto->setLabel("Enter restaurant's name");
34         $name_resto->addValidator(new PresenceOf([
35             'message'=>'Restaurant\'s name required'
36         ]));
37     }
```

4.9 Restaurant Form

```
app > form > RestaurantsForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\Text;
8  use Phalcon\Forms\Element\TextArea;
9  use Phalcon\Forms\Element\File;
10 use Phalcon\Forms\Element\Submit;
11
12 use Phalcon\Tag;
13
14 // validation
15 use Phalcon\Validation;
16 use Phalcon\Validation\Validator\PresenceOf;
17 use Phalcon\Validation\Validator\Alpha;
18 use Phalcon\Validation\Validator\StringLength;
19 use Phalcon\Validation\Validator\InclusionIn;
20
21
22 class RestaurantsForm extends BaseForm {
23     public function initialize(){
24
25         // ----- restaurant's name -----
26
27         $name_resto = new Text( 'resto_name',
28             [
29                 'placeholder' => 'Enter Restaurant\'s Name',
30                 'class' => 'form-control'
31             ]
32         );
33         $name_resto->setLabel("Enter restaurant's name");
34         $name_resto->addValidator(new PresenceOf([
35             'message'=>'Restaurant\'s name required'
36         ]));
37
38         // ----- restaurant's phone number -----
```

4.10 Reviews Form

```
app > form > ReviewsForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\Text;
8  use Phalcon\Forms\Element\Hidden;
9  use Phalcon\Forms\Element\TextArea;
10 use Phalcon\Forms\Element\Submit;
11
12 use Phalcon\Validation;
13 use Phalcon\Validation\Validator\PresenceOf;
14
15 use Phalcon\Tag;
16
17 class ReviewsForm extends BaseForm {
18     public function initialize(){
19         //$id_user = new Hidden('user_id');
20
21         // alias input
22         $name = new Text('alias',[
23             ]);
24         $name->addValidator( new PresenceOf(['message' => 'You should fill this field.']));
25
26         // id article
27         $id_resto = new Hidden('resto_id');
28         $id_resto->addValidator( new PresenceOf(['message' => 'Restaurant is not registered']));
29
30         // content of comment
31         $comment = new TextArea('review_contains');
32         $comment->addValidator( new PresenceOf(['message' => 'You should fill this field.']));
33
34         $rate = new Text('rating');
35         $rate->addValidator( new PresenceOf(['message' => 'You should fill this field.']));
36
37         $submit = new Submit('reviews',
38             [
39                 'value' => 'Submit',
40                 'class' => 'btn btn-primary'
41             ]);
42     }
```

4.11 Signup Form

```
app > form > SignupForm.php
1  <?php
2  Namespace App\Forms;
3  use App\Forms\BaseForm;
4  use Phalcon\Forms\Element\Text;
5  use Phalcon\Forms\Element\Radio;
6  use Phalcon\Forms\Element\Email;
7  use Phalcon\Forms\Element>Password;
8  use Phalcon\Forms\Element\Submit;
9
10 // VALIDATION
11 use Phalcon\Validation;
12 use Phalcon\Validation\Validator\PresenceOf;
13 use Phalcon\Validation\Validator\Regex;
14 use Phalcon\Validation\Validator\Alnum;
15 use Phalcon\Validation\Validator\StringLength;
16 use Phalcon\Validation\Validator\Confirmation;
17
18
19 use Phalcon\Tag;
20
21 class SignupForm extends BaseForm {
22     public function initialize(){
23         // ----- name -----
24         $name = new Text('name',
25             [
26                 "placeholder" => "Your Name",
27                 'class' => 'form-control',
28                 'value' => 'Anonymous'
29             ]
30         );
31         $name->addValidator(new StringLength(
32             [
33                 'min' => 3,
34                 'max' => 50,
35                 'messageMinimum' => 'Name must be at least 3 characters',
36                 'messageMaximum' => 'Name must be less than 50 characters'
37             ]
38         ));
39         $name->setLabel('What is your name?');
40     }
```

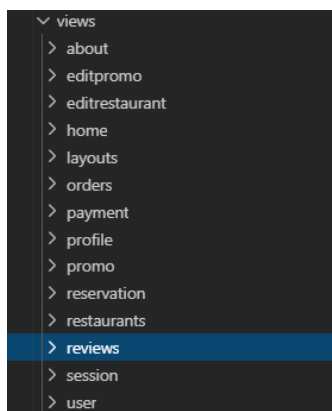
4.12 Upload Form

```
app > form > UploadForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\File;
8  use Phalcon\Forms\Element\Submit;
9
10 use Phalcon\Tag;
11
12 // validation
13 use Phalcon\Validation;
14 use Phalcon\Validation\Validator\File as FileValidator;
15 use Phalcon\Validation\Validator\PresenceOf;
16
17
18 class UploadForm extends BaseForm {
19     public function initialize(){
20         $photo = new File ('photo');
21         $photo->addValidator(
22             new FileValidator([
23                 "maxSize"           => "2M",
24                 "messageSize"       => ":field exceeds the max filesize (:max)",
25                 "allowedTypes"      => [
26                     "image/jpeg",
27                     "image/png",
28                 ],
29                 "messageType"       => "Allowed file types are :types",
30                 "maxResolution"    => "800x600",
31                 "messageMaxResolution" => "Max resolution of :field is :max",
32             ])
33         );
34
35         $this->add($photo);
36         $this->add(new Submit('Save'));
37     }
38 }
```

4.13 Validate Form

```
app > form > ValidateForm.php
1  <?php
2  Namespace App\Forms;
3
4  use App\Forms\BaseForm;
5
6  use Phalcon\Forms\Form;
7  use Phalcon\Forms\Element\Text;
8  use Phalcon\Forms\Element>Password;
9  use Phalcon\Forms\Element\Check;
10 use Phalcon\Forms\Element\Submit;
11 use Phalcon\Forms\Element\Hidden;
12
13 use Phalcon\Validation;
14 use Phalcon\Validation\Validator\Alnum;
15
16 use Phalcon\Tag;
17
18 class ValidateForm extends BaseForm {
19     public function initialize(){
20         $cek = new Check('remember',
21             [
22                 'value' => 1,
23             ]
24         );
25         $cek->setLabel('Remember me');
26
27         $this->add($username);
28         $this->add($password);
29         $this->add($id_user);
30         $this->add($cek);
31         $this->add($submit);
32     }
33 }
```

5. View



Terdapat 14 folder view pada pengerjaan project. Berikut adalah detailnya :

```

app > views > layouts > base.volt
1  <!DOCTYPE html>
2  <html>
3      <head>
4          {% include 'layouts/header.volt' %}
5          <title>{% block title %}{% endblock %} - My Webpage</title>
6      </head>
7      <body>
8          <nav>
9              <div class="logo"><a href="{{ url('') }}"></a></div>
10             <div class="nav-cont-left">
11                 <ul>
12                     <li><a href="{{ url('') }}">Home</a></li>
13                     <li>Restaurant
14                         <ul>
15                             {% if session.get('auth')['role'] == 1 %}
16                             <li><a href="{{ url('editresto') }}">List</a></li>
17                             {% else %}
18                             <li><a href="{{ url('restaurant-list') }}">List</a></li>
19                             {% endif %}
20                             {% if session.has('auth') %}
21                             <li><a href="{{ url('restaurants') }}">Register</a></li>
22                             {% endif %}
23                         </ul>
24                     </li>
25                     <li>Promo
26                         <ul style="margin-left:110px;">
27                             {% if session.get('auth')['role'] == 1 %}
28                             <li><a href="{{ url('editpromo') }}">List Promo</a></li>
29                             <li><a href="{{ url('menus') }}">Add Menu</a></li>
30                             {% endif %}
31                             <li><a href="{{ url('promo') }}">Promo</a></li>
32                         </ul>
33                     </li>
34                     {% if session.get('auth')['role'] == 1 %}
35                     <li>Validate
36                         <ul style="margin-left:190px;">
37                             <li><a href="{{ url('users') }}">User</a></li>
38                             <li><a href="{{ url('orders') }}">Order</a></li>
39                             <li><a href="{{ url('reservations') }}">Reservation</a></li>
40                         </ul>
41                     </li>

```

```

app > views > orders > create.volt
1  [% extends "layouts\base.volt" %]
2
3  {% block title %}Order Form{% endblock %}
4
5  {% block content %}
6
7  <div style="background-image: url('img/background1.png'); background-repeat: no-repeat;
8      background-attachment: fixed;
9      background-size: cover;">
10
11  <div class='notif-blok'>
12  {% if(error!='') %}
13      {{ this.flash.error(error) }}
14  {% endif %}
15  {% if(notif!='') %}
16      {{ this.flash.success(notif) }}
17  {% endif %}
18  </div>
19
20
21  <div class="wrapper">
22  <form action="order" method='POST' enctype='multipart/form-data' id="wizard">
23      <h4></h4>
24      <section>
25          <div class="inner">
26              <div class="image-holder">
27                  
28              </div>
29              <div class="form-content">
30                  <div class="form-inner">
31                      <div class="form-header">
32                          <h3>Order Form</h3>
33                          <p><br></p>
34                      </div>
35                      <div class="form-row">
36                          <label for='resto_name'>{{ form.getLabel('resto_name') }}</label>
37                          <div class="out-flash">{{ this.flash.outputMessage('errorf', '**~messages['resto_name'] ) }}
38                          {{ form.render('resto_name') }}
39                      </div>

```



```

app > views > payment > create.volt
1  [% extends "layouts\base.volt" %]
2
3  {% block title %}Payment Form{% endblock %}
4
5  {% block content %}
6
7  <div style="background-image: url('img/background1.png'); background-repeat: no-repeat;
8      background-attachment: fixed;
9      background-size: cover;">
10
11  <div class='notif-blok'>
12      {% if(error!='') %}
13          {{ this.flash.error(error) }}
14      {% endif %}
15      {% if(notif!='') %}
16          {{ this.flash.success(notif) }}
17      {% endif %}
18  </div>
19
20
21  <div class="wrapper">
22      <form action="payment" method='POST' enctype='multipart/form-data' id="wizard">
23          <h4></h4>
24          <section>
25              <div class="inner">
26                  <div class="image-holder">
27                      
28                  </div>
29                  <div class="form-content">
30                      <div class="form-inner">
31                          <div class="form-header">
32                              <h3>Payment Form</h3>
33                              <p><br></p>
34                          </div>
35                          <div class="form-row">
36                              <label for='type'>{{ form.getLabel('type') }}</label>
37                              <div class="out-flash">{{ this.flash.outputMessage('errorf', '*~messages['type']') }}
38                                  {{ form.render('type') }}
39                              </div>

```

```

app > views > promo > create.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Add Promo{% endblock %}
4
5  {% block content %}
6  <div style="background-image: url('img/background1.png'); background-repeat: no-repeat;
7    background-attachment: fixed;
8    background-size: cover;">
9
10 <div style="height:20px; margin-top:65px;">
11   <div class='notif-blok'>
12     {% if notif !='' %}
13     {{ this.flash.success(notif) }}
14     {% endif %}
15     {% if error !='' %}
16     {{ this.flash.error(error) }}
17     {% endif %}
18   </div>
19 </div>
20
21 <div class="wrapper" style="margin-top:-3%;">
22   <form action="promo" method='POST' enctype='multipart/form-data' id="wizard">
23     <!-- SECTION 1 -->
24     <h4></h4>
25     <section>
26       <div class="inner">
27         <div class="image-holder">
28           
29         </div>
30         <div class="form-content">
31           <div class="form-inner">
32             <div class="form-header" style="height:20%">
33               <h3>Add Promo</h3>
34             </div>
35             <div class="form-row">
36               <label for='promo_name'>{{ form.getLabel('promo_name') }}</label>
37               <div class="out-flash">{{ this.flash.outputMessage('errorf', ''~messages['promo_name']) }}
38                 {{ form.render('promo_name') }}
39             </div>

```

```

app > views > reservation > ≡ create.volt
1  [% extends "layouts\base.volt" %]
2
3  {% block title %}Order Form{% endblock %}
4
5  {% block content %}
6
7  <div style="background-image: url('img/background1.png'); background-repeat: no-repeat;
8      background-attachment: fixed;
9      background-size: cover;">
10
11  <div class='notif-blok'>
12  <div>
13      {% if(error!='') %}
14          {{ this.flash.error(error) }}
15      {% endif %}
16      {% if(notif!='') %}
17          {{ this.flash.success(notif) }}
18      {% endif %}
19  </div>
20
21  <div class="wrapper">
22  <div>
23      <form action="reservation" method='POST' enctype='multipart/form-data' id="wizard">
24      <div>
25          <h4></h4>
26          <section>
27          <div class="inner">
28          <div class="image-holder">
29              
30          </div>
31          <div class="form-content">
32          <div class="form-inner">
33          <div class="form-header">
34              <h3>Reservation Form</h3>
35              <p></p>
36          </div>
37          <div style="margin-top:-3vw;">
38          <div class="form-row">
39              <label for='resto_name'>{{ form.getLabel('resto_name') }}</label>
40              <div class="out-flash">{{ this.flash.outputMessage('errorf', '**~messages['resto_name']

```

```

app > views > restaurants > create.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Register Restaurant{% endblock %}
4
5  {% block content %}
6  <div style="background-image: url('img/background1.png'); background-repeat: no-repeat;
7    background-attachment: fixed;
8    background-size: cover;">
9
10 <div style="height:20px; margin-top:65px;">
11   <div class='notif-blok'>
12     {% if notif !='' %}
13     {{ this.flash.success(notif) }}
14     {% endif %}
15     {% if error !='' %}
16     {{ this.flash.error(error) }}
17     {% endif %}
18   </div>
19 </div>
20
21 <div class="wrapper" style="margin-top:-5%;">
22   <form action="restaurants" method='POST' enctype='multipart/form-data' id="wizard">
23     <!-- SECTION 1 -->
24     <h4></h4>
25     <section>
26       <div class="inner">
27         <div class="image-holder">
28           
29         </div>
30         <div class="form-content">
31           <div class="form-inner">
32             <div class="form-header">
33               <h3>Registration for Restaurant</h3>
34               <p>~ About Identity ~</p>
35             </div>
36             <div class="form-row">
37               <label for='resto_name'>{{ form.getLabel('resto_name') }}</label>
38               <div class="out-flash">{{ this.flash.outputMessage('errorf', '~messages['resto_name'] ) }}
39               {{ form.render('resto_name') }}
40             </div>

```

```

app > views > reviews > ≡ create.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Review and Rating{% endblock %}
4
5  {% block content %}
6
7  <div style="height:20px; margin-top:65px;">
8      <div class='notif-blok'>
9          {% if notif !='' %}
10             {{ this.flash.success(notif) }}
11          {% endif %}
12          {% if error !='' %}
13             {{ this.flash.error(error) }}
14          {% endif %}
15      </div>
16  </div>
17
18  <div class="wrapper">
19      <form action="reviews" method='POST' enctype='multipart/form-data' id="wizard">
20          <h4></h4>
21          <section>
22              <div class="inner">
23                  <div class="image-holder">
24                      
25                  </div>
26                  <div class="form-content">
27                      <div class="form-inner">
28                          <div class="form-header">
29                              <h3>Review Form</h3>
30                              <p><br></p>
31                          </div>
32                          <div class="form-row" style="margin-top:-3vw;">
33                              <label for='rating'>{{ form.getLabel('rating') }}</label>
34                              <div class="out-flash">{{ this.flash.outputMessage('errorf', '**~messages['rating']') }}</div>
35                              <p>{{ form.render('rating') }}</p>
36                          </div>
37                          <div class="form-row" id='big'>
38                              <label for='alias'>{{ form.getLabel('alias') }}</label>
39                              <div class="out-flash">{{ this.flash.outputMessage('errorf', '**~messages['alias']') }}</div>
40                              <p>{{ form.render('alias') }}</p>
41                          </div>

```

```

app > views > session > create.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Login Page{% endblock %}
4
5  {% block content %}
6
7  <div style="background-image: url('img/background1.png'); background-repeat: no-repeat;
8      background-attachment: fixed;
9      background-size: cover;height=100vw;">
10
11  <div class="row-centered">
12      <div class="card" style="display: relative; z-index: 1;">
13          <div class="notif-blok">
14              {% if(message!='') %}
15                  {{ this.flash.error(message) }}
16              {% endif %}
17          </div>
18      </div>
19      <div class="col-md-6">
20          {{ form.startForm() }}
21          <div class="form-group">
22              {{ form.render('username') }}
23          </div>
24          <div class="form-group">
25              {{ form.render('password') }}
26          </div>
27          <div class="form-row" style="display: block;">
28              {{ form.render('Login') }}
29          </div>
30          <div class="form-row" style="display: block;">
31              {{ form.render('remember') }}
32              {{ form.getLabel('remember') }}
33          <div class="regis">
34              <br><p><br>Don't have any account ?
35              <a href="{{url('register')}}">Register</a></p>
36          </div>
37      </div>
38      {{ form.endForm() }}
39      <div class="bvg">
40          
41

```

```

app > views > user > create.volt
1  [% extends "layouts\base.volt" %]
2
3  {% block title %}Register Page{% endblock %}
4
5  {% block content %}
6
7  <div style="background-image: url('img/background1.png'); background-repeat: no-repeat;
8      background-attachment: fixed;
9      background-size: cover;">
10
11  <div style="height:20px; margin-top:65px;">
12      <div class='notif-blok'>
13          {% if notif !='' %}
14              {{ this.flash.success(notif) }}
15          {% endif %}
16          {% if error !='' %}
17              {{ this.flash.error(error) }}
18          {% endif %}
19      </div>
20  </div>
21
22  <div class="wrapper" style="margin-top:-2%;">
23      <form action="register" method="POST" id="wizard">
24          <!-- SECTION 1 -->
25          <h4></h4>
26          <section>
27              <div class="inner">
28                  <div class="image-holder">
29                      
30                  </div>
31                  <div class="form-content">
32                      <div class="form-inner">
33                          <div class="form-header">
34                              <h3>Registration to be Member</h3>
35                              <p>~ About Your Identity ~</p>
36                          </div>
37
38                          <div class="form-row">
39                              {{ form.startForm() }}
40                              <label for='name'>{{ form.getLabel('name') }}</label>
41                              {{ form.render('name') }}

```

```

app > views > editpromo > edit.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Edit Cat's Information{% endblock %}
4
5  {% block content %}
6
7  <div class='notif-blok'>
8      {% if(error!='') %}
9          {{ this.flash.error(error) }}
10     {% endif %}
11     {% if(notif!='') %}
12         {{ this.flash.success(notif) }}
13     {% endif %}
14 </div>
15
16
17 <div class='editcat-box' style="width:50vw;height:50vw;">
18     <h1>Edit Promo's Information</h1>
19     <img src={{promo.photo_promo}} height=100 px;>
20
21     {{ form.startForm() }}
22     {{ form.render('id_promo') }}
23     <div style="color:black;">
24
25         <label for='promo_name'>{{form.getLabel('promo_name')}} </label>
26         {{ form.render('promo_name') }}
27
28         <label for='category'>{{form.getLabel('category')}} </label>
29         {{ form.render('category') }}
30
31         {{ form.render('Update') }}
32     {{ form.endForm() }}
33 </div>
34 </div>
35
36 {% endblock %}

```



```

app > views > editrestaurant > edit.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Edit Cat's Information{% endblock %}
4
5  {% block content %}
6
7  <div class='notif-blok'>
8      {% if(error!='') %}
9          {{ this.flash.error(error) }}
10     {% endif %}
11     {% if(notif!='') %}
12         {{ this.flash.success(notif) }}
13     {% endif %}
14 </div>
15
16
17 <div class='editcat-box' style="width:50vw;height:50vw;">
18 <h1>Edit Restaurant's Information</h1>
19 <img src={{restaurant.photo}} height=150 px>
20
21 {{ form.startForm() }}
22     {{ form.render('id_resto') }}
23     <div style="color:black;">
24         <label for='resto_name'>{{form.getLabel('resto_name')}} </label>
25         {{ form.render('resto_name') }}
26
27         <label for='phone_num'>{{form.getLabel('phone_num')}} </label>
28         {{ form.render('phone_num') }}
29
30         <label for='location'>{{form.getLabel('location')}} </label>
31         {{ form.render('location') }}
32
33         <label for='omzet'>{{form.getLabel('omzet')}} </label>
34         {{ form.render('omzet') }}
35
36         <label for='menu_name'>{{form.getLabel('menu_name')}} </label>
37         {{ form.render('menu_name') }}
38
39         <label for='price'>{{form.getLabel('price')}} </label>
40         {{ form.render('price') }}
41     </div>

```

app > views > layouts > footer.volt

```
1
2 <script src="{{ url("js/jquery.js") }}"></script>
3 <script src="{{ url("js/jquery-ui.js") }}"></script>
4 <script src="{{ url("js/bootstrap.js") }}"></script>
5 <script src="{{ url("js/popper.js") }}"></script>
6 <script src="{{ url("js/mdb.min.js") }}"></script>
7 <script src="{{ url("js/jquery.steps.js") }}"></script>
8 <script src="{{ url("js/script.js") }}"></script>
9 <script src="{{ url("js/script1.js") }}"></script>
10 <script src="{{ url("js/addons/datatables.min.js") }}"></script>
11
12 <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD2IXRLYl7bxjKUU6UaVlYBTCU04YvR5fQ&callback=myMap"></s
13
14
```

app > views > home > forbidden.volt

```
1 {% extends "layouts\base.volt" %}
2
3 {%block content %}
4 <div style="margin:200px;">
5     <h1 class='alert-danger'>You don't have permission to access this page.</h1>
6     <a href='./'>Back to home</a>
7 </div>
8
9 {% endblock %}
```

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="https://unpkg.com/font-awesome@4.7.0/css/font-awesome.min.css" rel="stylesheet" />

<link href="https://fonts.googleapis.com/css?family=Josefin+Slab:400,700" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Courgette|EB+Garamond|Charmonman" rel="stylesheet">

<link href="{{ url("css/bootstrap.css") }}" rel="stylesheet">

<link href="{{ url("css/ui.css") }}" rel="stylesheet">

<link href="{{ url("css/style.css") }}" rel="stylesheet">

<link href="{{ url("css/style1.css") }}" rel="stylesheet">

<link href="{{ url("css/style2.css") }}" rel="stylesheet">

<link href="{{ url("css/addons/datatables.min.css") }}" rel="stylesheet">

<link href="{{ url("css/mdb.min.css") }}" rel="stylesheet">

```

app > views > home > index.volt
1  {% extends "layouts\base.volt" %}
2
3
4  {% block title %} Halaman Utama Saya {% endblock %}
5
6  {% block content %}
7  <div class="home" style="padding-bottom: 0 !important; padding-top: 0 !important; margin-bottom:0;">
8  <div style="background-image: url('img/background_card.jpg');background-size: cover; padding-bottom:50%;" height=1000
9
10     <div id="home-slide">
11     
12
13     <div class="card centered" style="top:25%;margin-left:15vw;width:42vw;height:28vw;opacity:0.9;">
14         <h1 class="centered5" style="top:30%;">
15             <a href="" class="typewrite" data-period="2000" data-type='[ "SIKULITS" ]'>
16                 <span class="wrap"></span>
17             </a>
18         </h1>
19         <h1 class="centered5" style="top:43%;">
20             <a href="" class="typewrite" data-period="4000" data-type='[ "Sistem Informasi", "Kuliner", "ITS" ]'>
21                 <span class="wrap"></span>
22             </a>
23         </h1>
24         <div class="row" style="margin-left:5vw;margin-top:5vw;">
25             <div class="col-md-6"><a href="{{ url("register") }}"><button>Sign Up</button></a></div>
26             <div class="col-md-6">
27                 {% if session.has('auth') %}
28                 <a href="{{ url("restaurant-list") }}"><button>Resto</button></a></div>
29                 {% else %}
30                 <a href="{{ url("login") }}"><button>Login</button></a></div>
31                 {% endif %}
32             </div>
33         </div>
34     </div>
35 </div>
36 </div>

```

```

app > views > promo > list.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}List Promo{% endblock %}
4
5  {% block content %}
6
7  <div class='notif-blok'>
8      {% if success != '' %}
9          {{ this.flash.success(success) }}
10         {% endif %}
11         {% if error != '' %}
12             {{ this.flash.error(error) }}
13         {% endif %}
14     </div>
15
16     <div class="container-fluid">
17         {% if count > 0 %}
18             <div id="myCarousel" class="carousel slide" data-ride="carousel">
19                 <div class="carousel-inner row w-100 mx-auto 0" role="listbox" style="width:100%; height:40vw">
20                     {% for p in promos %}
21                         {% if (loop.first) %}
22                             <div class="carousel-item col-md-4 active">
23                                 {% else %}
24                                     <div class="carousel-item col-md-4">
25                                         {% endif %}
26                                         <div class="card h-75" style="background-image: url('img/background_card.jpg');background-size: cover;">
27                                             <div class="card-body">
28                                                 <img src={{p.photo_promo}} style="height:30%;display: block; margin-left: auto; margin-right: auto;">
29                                                 <h4 class="card-title" align="center">{{p.promo_name}}</h4>
30                                                 <div>
31                                                     {{ 'Category promo : ' ~p.category~ ' - in ' ~p.available_in }}<br/>
32                                                     {{ 'Available start: ' ~p.available_start }} <br/>
33                                                     {{ 'Description: ' ~p.description_promo }} </br>
34                                                 </div>
35                                                 <button type='button' class='ad-btn btn btn-info' id={{p.id_promo}} value={{p.id_promo}} data-toggle=
36                                                 <p class="card-text">
37                                                     <small class="text-muted">Last updated {{ p.last_up }}</small>
38                                                 </p>
39                                         </div>

```

app > views > restaurants > list.volt

```
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}List Promo{% endblock %}
4
5  {% block content %}
6
7  <div class='notif-blok'>
8      {% if success != '' %}
9      {{ this.flash.success(success) }}
10     {% endif %}
11     {% if error != '' %}
12     {{ this.flash.error(error) }}
13     {% endif %}
14 </div>
15
16 <div class="container-fluid">
17     {% if count > 0 %}
18     <div id="myCarousel" class="carousel slide" data-ride="carousel">
19         <div class="carousel-inner row w-100 mx-auto 0" role="listbox" style="width:100%; height:40vw">
20             {% for r in restos %}
21                 {% if (loop.first) %}
22                 <div class="carousel-item col-md-4 active">
23                     {% else %}
24                     <div class="carousel-item col-md-4">
25                         {% endif %}
26                         {% set idx = loop.index %}
27                         {% set i = 1 %}
28                         <div class="card h-75">
29                             <div class="card-body" style="background-image: url('img/background_card.jpg');background-size: cover;">
30                                 <img src={{r.photo}} style="height:30%;display: block; margin-left: auto; margin-right: auto;">
31                                 <h4 class="card-title" align="center">{{r.resto_name}}</h4>
32                                 <div>
33                                     {{ 'Location : ' ~r.location }}<br/>
34                                     {{ 'Phone : ' ~r.phone_num }}<br>
35                                 </div>
36                                 <a href='menus?id_resto={{r.id_resto}}'><button type='button' class='ad-btn btn btn-info' data-id={{r
37                                 <a href='reviews?id_resto={{r.id_resto}}'><button type='button' class='ad-btn btn btn-info' data-id={{
38                                 <!--<button type='button' class='ad-btn btn btn-info' data-toggle='modal' data-target='#review' style
39                             </div>
40                         </div>
41                     </div>
```

app > views > home > show404.volt

```
1  {% extends "layouts\base.volt" %}
2
3  {%block content %}
4  <div style="margin:200px;">
5      <h1 class='alert-danger'>Page not found.</h1>
6      <a href='.'>Back to home</a>
7  </div>
8
9  {% endblock %}
```

```

app > views > editpromo > ≡ show.volt
2
3  {% block title %}Promo's List{% endblock %}
4
5  {% block content %}
6
7  <div class="container-fluid">
8    <div class="card" style="height:0.5vw; width:100vw; display:relative; z-index:1; margin-left:-16vw">
9      <div class='notif-blok'>
10        {% if success != '' %}
11        {{ this.flash.success(success) }}
12        {% endif %}
13        {% if error != '' %}
14        {{ this.flash.error(error) }}
15        {% endif %}
16      </div>
17    </div>
18    {% if countpromo > 0 %}
19    <div id="myCarousel" class="carousel slide" data-ride="carousel">
20      <div class="carousel-inner row w-100 mx-auto 0" role="listbox" style=" width:100%; height:45vw">
21        {% for c in promos %}
22          {% if (loop.first) %}
23          <div class="carousel-item col-md-4 active">
24            {% else %}
25            <div class="carousel-item col-md-4">
26              {% endif %}
27            <div class="card h-75">
28              <div class="card-body" style="background-image: url('img/background_card.jpg');background-size: cover;">
29                {% if c.last_up == null %}
30                {{ this.flash.outputMessage('errorf', ' (NEW!)') }}
31                {% endif %}
32                <img src={{c.photo_promo}} height=200px>
33                <h4 class="card-title">{{c.promo_name}}</h4>
34                <div>
35                  {{ 'Category : ' ~c.category }}<br/>
36                  {{ 'Description : ' ~c.description_promo }}
37                </div>
38              </div>
39              <form method='GET' action='editpromo'>
40                <input type='hidden' name='id_promo' value={{c.id_promo}} >
41                <div style='display:inline-block'>
42                  <button type='button' class='btn btn-danger' id={{c.id_promo}} value={{c.id_promo}} data-t

```

```

app > views > editrestaurant > show.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Restaurant's List{% endblock %}
4
5  {% block content %}
6
7  <div class="container-fluid">
8      <div class="card" style="height:0.5vw; width:100vw; display:relative; z-index:1; margin-left:-16vw">
9          <div class='notif-blok'>
10             {% if success != '' %}
11             {{ this.flash.success(success) }}
12             {% endif %}
13             {% if error != '' %}
14             {{ this.flash.error(error) }}
15             {% endif %}
16          </div>
17      </div>
18      {% if countresto > 0 %}
19          <div id="myCarousel" class="carousel slide" data-ride="carousel">
20              <div class="carousel-inner row w-100 mx-auto 0" role="listbox" style=" width:100%; height:45vw">
21                  {% for c in restos %}
22                      {% if (loop.first) %}
23                      <div class="carousel-item col-md-4 active">
24                          {% else %}
25                          <div class="carousel-item col-md-4">
26                          {% endif %}
27                      <div class="card h-75">
28                          <div class="card-body" style="background-image: url('img/background_card.jpg');background-size: cover;">
29                              {% if c.date_register == null %}
30                              {{ this.flash.outputMessage('errorf', ' (NEW!)') }}
31                              {% endif %}
32                              <img src={{c.photo}} height=200px>
33                              <h4 class="card-title">{{c.resto_name}}</h4>
34                              <div>
35                                  {{ 'Location : ' ~c.location }}<br/>
36                                  {{ 'Phone : ' ~c.phone_num }}
37                              </div>
38
39                              <form method='GET' action='editresto'>
40                                  <input type='hidden' name='id_resto' value={{c.id_resto}} >
41                                  <div style='display:inline-block'>

```

```

app > views > profile > show.volt
1  [% extends "layouts\base.volt" %]
2
3  {% block title %}Profile{% endblock %}
4
5  {% block content %}
6  <div class="row" style="margin-left:20vw; margin-top:-285px">
7  <div class="col-md-6">
8  <div class='notif-blok'>
9      {% if(messages['warning']!= '') %}
10         {{ this.flash.outputMessage('notice', messages['warning']) }}
11     {% endif %}
12     {% if(messages['error']!= '') %}
13         {{ this.flash.error(messages['error']) }}
14     {% endif %}
15     {% if(messages['success']!= '') %}
16         {{ this.flash.success(messages['success']) }}
17     {% endif %}
18 </div>
19
20 <div class='profile-box'>
21     <h1>YOUR PROFILE</h1>
22     <div class='container'>
23         <div>
24             <a href='data:image/png;base64,{{user.photo_user}}{{"/download="}} {{this.session.get('auth')['username']}}'>
25                 <img width=300px src=data:image/png;base64,{{user.photo_user}}>
26             </a>
27
28             
30     </div>
31     <div class="modal" id="myModal" role="dialog">
32         <div class="modal-dialog">
33
34             <!-- Modal content-->
35             <div class="modal-content">
36                 <div class="modal-header">
37                     <h4 class="modal-title">Upload Profile Picture</h4>
38                     <button type="button" class="close" data-dismiss="modal">&times;</button>
39                 </div>
40                 <div class="modal-body">

```



```

app > views > orders > ≡ validate.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Validate Order{% endblock %}
4
5  {% block content %}
6  <div class='container' id='donation-list' style="margin:100px 200px">
7    <h1>Unvalidate Order List</h1>
8
9    {% if count <= 0 %}
10     <h4 style="color:red;">All orders already validated</h4>
11
12    {% else %}
13
14     {% set iter = 1 %}
15     <table class="table">
16       <thead>
17         <th>No</th>
18         <th>Order Id</th>
19         <th>User Id</th>
20         <th>Resto Id</th>
21         <th>Menu Id</th>
22         <th>Amount</th>
23         <th>Change Status of Validation</th>
24       </thead>
25       {% for u in orders %}
26         <tr>
27           <td>{{iter}}</td>
28           <td>{{u.idorder}}</td>
29           <td>{{u.iduser}}</td>
30           <td>{{u.idmenu}}</td>
31           <td>{{u.idresto}}</td>
32           <td>{{u.amount}}</td>
33           <td><button>Change</button></td>
34         </tr>
35         {% set iter+=1 %}
36       {% endfor %}
37     </table>
38   {% endif %}
39 </div>
40

```

```

app > views > reservation > ≡ validate.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Validate Reservation{% endblock %}
4
5  {% block content %}
6  <div class='container' id='donation-list' style="margin:100px 200px">
7    <h1>Unvalidate Reservation List</h1>
8
9    {% if count <= 0 %}
10     <h4 style="color:red;">All reservation already validated</h4>
11
12    {% else %}
13
14     {% set iter = 1 %}
15     <table class="table">
16       <thead>
17         <th>No</th>
18         <th>Reservation Id</th>
19         <th>User Id</th>
20         <th>Id Resto</th>
21         <th>Amount</th>
22         <th>Change Status of Validation</th>
23       </thead>
24       {% for u in reserves %}
25         <tr>
26           <td>{{iter}}</td>
27           <td>{{u.idreserve}}</td>
28           <td>{{u.userid}}</td>
29           <td>{{u.restoid}}</td>
30           <td>{{u.amount}}</td>
31           <td><button>Change</button></td>
32         </tr>
33         {% set iter+=1 %}
34       {% endfor %}
35     </table>
36   {% endif %}
37 </div>
38
39 {% endblock %}

```

```

app > views > user > ≡ validate.volt
1  {% extends "layouts\base.volt" %}
2
3  {% block title %}Validate Member{% endblock %}
4
5  {% block content %}
6  <div class='container' id='donation-list' style="margin:100px 200px">
7    <h1>Unvalidate User List</h1>
8
9    {% if count <= 0 %}
10     <h4 style="color:red;">All users already validated</h4>
11
12    {% else %}
13
14     {% set iter = 1 %}
15     <table class="table">
16       <thead>
17         <th>No</th>
18         <th>Username</th>
19         <th>Name</th>
20         <th>Email</th>
21         <th>Phone</th>
22         <th>Change Status of Validation</th>
23       </thead>
24       {% for u in users %}
25         <tr>
26           <td>{{iter}}</td>
27           <td>{{u.username}}</td>
28           <td>{{u.name}}</td>
29           <td>{{u.email}}</td>
30           <td>{{u.phone}}</td>
31           <td><button>Change</button></input></td>
32         </tr>
33         {% set iter+=1 %}
34       {% endfor %}
35     </table>
36   {% endif %}
37 </div>
38
39 {% endblock %}

```