Login

$this->permission();

$this->data['page']['title'] = 'Login Page';

$this->data['form']['action\_login'] = base\_url('admin/auth');

$this->form\_validation->set\_rules('email', 'Email', 'trim|required|valid\_email');

$this->form\_validation->set\_rules('password', 'Password', 'trim|required');

if ($this->form\_validation->run() == false) {

$this->template->load('admin', 'user/login', $this->data);

} else {

// validasinya success

$this->\_login();

}

Method login

private function \_login()

{

$email = $this->input->post('email');

$this->db->where("email", $email);

$role = $this->data['page']['id'];

$role ? $this->db->where("role\_id", $role) : '';

$user = $this->db->get("tbl\_user")->row\_array();

// jika usernya ada

if ($user) {

// jika usernya aktif

if ($user['is\_active'] == 1) {

// cek password

$password = $this->input->post('password');

if (password\_verify($password, $user['password'])) {

$data = [

'time' => time() + (60 \* 60),

'email' => $user['email'],

'role\_id' => $user['role\_id']

];

$this->session->set\_userdata($data);

$this->session->set\_flashdata('notif', '<div class="alert alert-succes" role="alert">Succes Login!</div>');

if ($this->session->userdata('url')) {

redirect($this->session->userdata('url'));

}

$user['role\_id'] == 3 ? redirect("user") : redirect('admin/dashboard');

} else {

$this->session->set\_flashdata('message', '<div class="alert alert-danger" role="alert">Wrong password!</div>');

if ($this->session->userdata('email')) {

redirect('admin/auth/lock');

} else

redirect('admin/auth');

}

} else {

$this->session->set\_flashdata('message', '<div class="alert alert-danger" role="alert">This email has not been activated!</div>');

redirect('admin/auth');

}

} else {

$this->session->set\_flashdata('message', '<div class="alert alert-danger" role="alert">Email is not registered!</div>');

redirect('admin/auth');

}

}

Halaman admin menampilkan daftar aspirasi

$this->data['page']['title'] = 'Managemen Aspirasi Masyarakat';

$this->data['page']['description'] = 'Silahkan edit Atau Hapus jika ada kesalahan Komisi.!';

// $this->data['page']['before'] = ['url' => base\_url('admin/admin/role'), "title" => "Role Access"];

$this->data['page']['submenu'] = 'Daftar Aspirasi';

$this->data['all\_komisi'] = $this->db->get("web\_komisi")->result\_array();

$this->template->load('admin', 'aspirasi/admin', $this->data);

Untuk halaman htmlnya :

<!DOCTYPE html>

<html lang="en">

<?php

$this->load->view($thema\_load . 'element/template/head\_meta.php');

?>

<body class="">

<!-- sidebar -->

<?php $this->load->view($thema\_load . 'element/template/sidebar.php'); ?>

<!-- end sidebara -->

<?php

$this->load->view($thema\_load . 'element/template/navbar.php');

?>

<!-- isi content -->

<div class="card">

<div class="card-header card-header-primary card-header-icon">

<div class="card-icon">

<i class="material-icons">assignment</i>

</div>

<h4 class="card-title">Daftar Aspirasi Masyarakat</h4>

</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-hover">

<thead class="load">

<tr>

<th scope="col">#</th>

<th scope="col">User</th>

<th scope="col">Message</th>

<th scope="col">Komisi</th>

<th scope="col">Action</th>

</tr>

</thead>

<tbody class="daftar-aspirasi">

</tbody>

</table>

</div>

</div>

</div>

<!-- end isi -->

<?php

$this->load->view($thema\_load . 'element/template/footer.php');

?>

<?php

$this->load->view($thema\_load . 'element/template/fixed-setting.php');

?>

<div class="modal fade" id="edit" tabindex="-1" role="dialog" aria-labelledby="#modalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="modalLabel">Edit Aspirasi Dari User</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<form action="" method="post" id="formInput">

<div class="modal-body">

<div class="form-group mb-4">

<select class="form-control" data-style="btn btn-link" id="komisi\_id" name="komisi\_id">

<?php foreach ($all\_komisi as $row) : ?>

<option value="<?= $row['id\_komisi'] ?>"><?= $row['name'] ?></option>

<?php endforeach; ?>

</select>

</div>

<div class="form-group mt-5">

<label for="message">Pesan Aspirasi</label>

<textarea class="form-control" name="message" id="message" rows="2"></textarea>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

<button type="submit" class="btn btn-primary submit">Edit</button>

</div>

</form>

</div>

</div>

</div>

<script>

var dftAspirasi = $("tbody.daftar-aspirasi");

var tr = null;

</script>

<script src="<?= $thema\_folder ?>layout/aspirasi/admin.js"></script>

</body>

</html>

Halaman admin.js

var \_aspirasi = null;

var \_list = null;

dftAspirasi.on("click", ".mdl#delete", function() {

\_aspirasi = $("tr[id=" + $(this).data("id") + "]");

const url\_delete = $(this).data("url");

$(".modal#logout")

.find(".btn.url")

.attr("href", url\_delete)

.html("Delete");

$(".modal#logout")

.find(".modal-title")

.html("Jika Anda menghapus maka data yang berkaitan akan terhapus juga");

});

function tbodySort() {

$("tbody > tr")

.sort(function(a, b) {

return $("td.number", b).text() > +$("td.number", a).text();

})

.appendTo("tbody")

.find("th:first")

.text(function(index) {

return ++index + ".";

});

}

$("a.url.mdl").on("click", function(e) {

e.preventDefault();

var parents = $(this).parents(".modal-content");

let url = $(this).attr("href");

$.ajax({

url: url,

type: "DELETE",

dataType: "json",

beforeSend: () => {

parents.after("<div class='mdlProgresBarr'></div>");

$(".fullpage").addClass("tampil");

},

success: data => {

if (data.status) {

\_aspirasi.remove();

tbodySort();

}

message = data.message;

status = data.status;

},

error: function(e) {

console.log("error");

},

complete: function() {

$("head").append("<style>.mdlProgresBarr::before{width:100%;}</style>");

$(".mdlProgresBarr").remove();

$(".modal").modal("hide");

let css = status ? "success" : "danger";

let notif = `<div class="alert alert-${css}">

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<i class="material-icons">close</i>

</button>

<span>${message}</span>

</div>`;

$(".fullpage").hide("slow", function() {

$(".fullpage").remove();

});

$(".notif#ajax").html(notif);

$(".main-panel").animate(

{

scrollTop: $(".notif#ajax").offset().top

},

"slow"

);

}

});

});

dftAspirasi.on("click", ".mdl#edit", function() {

\_aspirasi = $("tr[id=" + $(this).data("id") + "]");

let url = baseUrl + "api/aspirasi?id\_aspirasi=" + $(this).data("id");

let komisi\_id = $(this).data("komisi");

let message = $(this).data("message");

$("#formInput").attr("action", url);

$("#formInput")

.find("option[value=" + komisi\_id + "]")

.attr("selected", true);

$("textarea[name=message]").val(message);

});

$().ready(function() {

$.ajax({

url: baseUrl + "api/aspirasi",

type: "get",

dataType: "json",

beforeSend: () => {

$("thead.load").after("<div class='mdlProgresBarr'></div>");

},

success: respon => {

if (respon.status) {

let Urut = 1;

respon.data.forEach(e => {

let tr = `

<tr id="${e.id\_aspirasi}">

<th scope="row">${Urut}</th>

<td name='username'>${e.username}</td>

<td name='message'>${e.message}</td>

<td name='komisi\_id'>${e.komisi}</td>

<td>

<a href="#" class="badge badge-primary mdl btn" data-komisi="${e.komisi\_id}" data-message="${e.message}" data-id="${e.id\_aspirasi}" data-toggle="modal" data-target=".modal#edit" id="edit">Edit</a>

<a href="" class="mdl badge badge-danger" id="delete" data-id="${e.id\_aspirasi}" data-toggle="modal" data-url="${baseUrl}api/aspirasi?id\_aspirasi=${e.id\_aspirasi}" data-target=".modal#logout">delete</a>

</td>

</tr>

`;

dftAspirasi.append(tr);

Urut++;

});

} else {

dftAspirasi.html(`<tr><td col=5>${respon.message}</td></tr>`);

}

},

error: e => {

console.log(e);

},

complete: () => {

$(".mdlProgresBarr").remove();

}

});

});

$("#formInput").on("submit", function(e) {

e.preventDefault();

var myForm = $(this);

let url = myForm.attr("action");

var data = $(this).serializeArray(); //serialize form inputs and pass them to php

$.ajax({

url: url,

type: "PUT",

data: data,

dataType: "json",

beforeSend: () => {

myForm.after("<div class='mdlProgresBarr'></div>");

$(".fullpage").addClass("tampil");

},

success: dt => {

let userId = $('input[name="user\_id"]').val();

let nameKomisi = $("select[name='komisi\_id'] option:selected").text();

if (dt.status) {

\_aspirasi

.children("td[name=komisi\_id]")

.text($("#komisi\_id option:selected").text());

\_aspirasi

.children("td[name=message]")

.text($("textarea[name=message]").val());

}

status = dt.status;

message = dt.message;

},

complete: function() {

$("head").append("<style>.mdlProgresBarr::before{width:100%;}</style>");

$(".mdlProgresBarr").remove();

$(".modal").modal("hide");

let css = status ? "success" : "danger";

let notif = `<div class="alert alert-${css}">

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<i class="material-icons">close</i>

</button>

<span>${message}</span>

</div>`;

$(".fullpage").hide("slow", function() {

$(".fullpage").remove();

});

$(".notif#ajax").html(notif);

$(".main-panel").animate(

{

scrollTop: $(".notif#ajax").offset().top

},

"slow"

);

}

});

});

Halaman admin, menampilkan daftar komisi

$this->data['page']['title'] = 'Mengolah Komisi';

$this->data['page']['description'] = 'Silahkan edit menu Atau Tambahkan Komisi.!';

// $this->data['page']['before'] = ['url' => base\_url('admin/admin/role'), "title" => "Role Access"];

$this->data['page']['submenu'] = 'Menu Komisi';

$validation = $this->form\_validation;

$komisi = $this->komisi\_m;

$this->data['all\_komisi'] = $komisi->getWhere()->result\_array();

$validation->set\_rules($komisi->getRules());

if ($validation->run() == false) {

$this->template->load('admin', 'komisi/home', $this->data);

} else {

$komisi->setPrimaryKey($komisi->getLastId());

$komisi->setFieldTable($this->post);

$komisi->add();

$ks = hasilCUD("Komisi Ditambahkan");

$type = $ks->status ? "success" : "danger";

$this->session->set\_flashdata("message","<div class='alert alert-$type' role='alert'> $ks->message</div>");

redirect(base\_url('admin/komisi/home'));

}

Home.php

<!DOCTYPE html>

<html lang="en">

<?php

$this->load->view($thema\_load . 'element/template/head\_meta.php');

?>

<body class="">

<!-- sidebar -->

<?php $this->load->view($thema\_load . 'element/template/sidebar.php'); ?>

<!-- end sidebara -->

<?php

$this->load->view($thema\_load . 'element/template/navbar.php');

?>

<!-- isi content -->

<div class="card">

<div class="card-header card-header-primary card-header-icon">

<div class="card-icon">

<i class="material-icons">assignment</i>

</div>

<h4 class="card-title">Daftar Komisi</h4>

</div>

<div class="card-body">

<a href="" data-url="<?= base\_url('admin/komisi/home'); ?>" id="addKomisi" class="btn btn-primary mb-3 mdl add" data-toggle="modal" data-target=".modal#komisi">Add New Komisi</a>

<div class="table-responsive">

<table class="table table-hover">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Nama</th>

<th scope="col">Action</th>

</tr>

</thead>

<tbody>

<?php $i = 1; ?>

<?php foreach ($all\_komisi as $row) : ?>

<tr>

<th scope="row"><?= $i; ?></th>

<td><?= $row['name']; ?></td>

<td>

<a href="<?= base\_url('admin/komisi/label/') . $row['id\_komisi']; ?>" data-id="<?= $row['id\_komisi'] ?>" class="badge badge-primary" id="label">Show Label</a>

<a href="" data-name="<?= $row['name'] ?>" data-url="<?= base\_url('admin/komisi/home/edit/') . $row['id\_komisi']; ?>" class="badge badge-success mdl" id="edit" data-toggle="modal" data-target=".modal#komisi">edit</a>

<a href="" class="mdl badge badge-danger" id="delete" data-toggle="modal" data-url="<?= base\_url('admin/komisi/home/delete/') . $row['id\_komisi']; ?>" data-target=".modal#logout">delete</a>

</td>

</tr>

<?php $i++; ?>

<?php endforeach; ?>

</tbody>

</table>

</div>

</div>

</div>

<!-- end isi -->

<?php

$this->load->view($thema\_load . 'element/template/footer.php');

?>

<?php

$this->load->view($thema\_load . 'element/template/fixed-setting.php');

?>

<!-- modal dan scrip code -->

<!-- Modal -->

<div class="modal fade" id="komisi" tabindex="-1" role="dialog" aria-labelledby="#modalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="modalLabel">Add New Menu</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<form action="<?= base\_url('admin/komisi'); ?>" method="post" id="formInput">

<div class="modal-body">

<div class="form-group">

<label for="#komisi" class="bmd-label-floating">Nama Komisi .\*</label>

<input type="text" class="form-control input name" id="komisi" required="true" name="name">

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

<button type="submit" class="btn btn-primary submit">Add</button>

</div>

</form>

</div>

</div>

</div>

<!-- jqery required -->

<!-- <script src="<?= $thema\_folder ?>assets/js/plugins/jquery.validate.min.js"></script> -->

<script src="<?= $thema\_folder ?>assets/js/plugins/jquery.validate.min.js"></script>

<script>

$('.mdl.add').on('click', function() {

const url\_add = $(this).data('url');

$('#formInput').attr('action', url\_add).parent().find("#modalLabel").html('Add Komisi');

$('#formInput').find("button.submit").html('Add Komisi');

$('#formInput').find(".input.name").val('').parent().removeClass('is-filled');

});

$('.mdl#edit').each(function(index) {

$(this).on('click', function() {

const val\_name = $(this).data('name');

const url\_edit = $(this).data('url');

$('#formInput').attr('action', url\_edit).parent().find("#modalLabel").html('Edit Komisi');

$('#formInput').find("button.submit").html('Edit Komisi');

$('#formInput').find(".input.name").val(val\_name).parent().addClass('is-filled');

});

})

$('.mdl#delete').each(function() {

$(this).on('click', function() {

const url\_delete = $(this).data('url');

console.log(url\_delete);

$(".modal#logout").find('.btn.url').attr('href', url\_delete).html("Delete");

$(".modal#logout").find(".modal-title").html("Jika Anda menghapus maka data yang berkaitan akan terhapus juga");

});

})

// validate modal

$(function() {

$("#formInput").validate({

rules: {

name: {

required: true,

minlength: 4

},

action: "required"

},

messages: {

name: {

required: "Please enter some data",

minlength: "Your data must be at least 4 characters"

},

action: "Please provide some data"

}

});

});

</script>

</body>

</html>

Menambahkan user ke komisi tertentu

$this->load->library('form\_validation');

$this->form\_validation->set\_rules("name", "Nama Lengkap", "required|min\_length[5]");

$this->form\_validation->set\_rules("email", "email", "required|is\_unique[tbl\_user.email]");

$this->form\_validation->set\_rules("no\_hp", "No Handphone", "required|numeric");

$this->form\_validation->set\_rules("alamat", "Alamat", "required");

$this->form\_validation->set\_rules("password", "Password", "required");

$this->form\_validation->set\_rules("role\_id", "role id", "required");

$this->form\_validation->set\_rules("dapil\_id", "Dapil", "required");

$this->form\_validation->set\_rules("komisi\_id", "Komisi", "required");

$this->form\_validation->set\_rules("jabatan", "jabatan", "required");

if ($this->form\_validation->run()) {

$post = $this->input->post();

$image = "default.png";

if (isset($\_FILES['image'])) {

$config['allowed\_types'] = 'gif|jpg|jpeg|png';

$config['max\_size'] = '3048';

$config['upload\_path'] = './assets/img/profile/';

$this->load->library('upload', $config);

if ($this->upload->do\_upload('image')) {

$image = $this->upload->data('file\_name');

/\*\*

\* Helper CI

\* Create thumb image with

\* \_img\_create\_thumbs({nama file image}, {Folder file gambar});

\*/

\_img\_create\_thumbs($image, "profile");

}

}

$tblUser = initTable("tbl\_user", "user");

$tblUser['lastkey'] = $tblUser['field'][$tblUser['key']];

$data = [

$tblUser['key'] => $tblUser['lastkey'],

"name" => $post['name'],

"email" => $post['email'],

"no\_hp" => $post['no\_hp'],

"alamat" => $post['alamat'],

"role\_id" => $this->input->post('role\_id') ? $this->input->post('role\_id') : 3,

"is\_active" => $this->input->post('is\_active') ? $post['is\_active'] : 0,

"image" => $image,

"password" => password\_hash($post['password'], PASSWORD\_DEFAULT)

];

$this->db->insert($tblUser['name'], $data);

$tbl = initTable("web\_komisi\_user", "kom\_u");

$this->db->insert($tbl['name'], [

$tbl['key'] => $tbl['field'][$tbl['key']],

"komisi\_id" => $this->input->post("komisi\_id"),

"dapil\_id" => $this->input->post("dapil\_id"),

"user\_id" => $data["id\_user"],

"jabatan" => $this->input->post("jabatan"),

]);

$respon = hasilCUD("Data User Ditambahkan");

$type = $respon->status ? "success" : "danger";

$this->session->set\_flashdata('message', '<div class="alert alert-' . $type . '" role="alert">' . $respon->message . '!</div>');

} else {

$this->session->set\_flashdata('message', '<div class="alert alert-danger" role="alert">Gagal..!</div>');

}

redirect("admin/komisi/user");

Untuk mengedit user komisi

$user = $this->db->get\_where("tbl\_user", ['id\_user' => $id])->row\_array();

if ($user) {

$this->load->library('form\_validation');

$this->form\_validation->set\_rules("name", "Nama Lengkap", "required|min\_length[5]");

$this->form\_validation->set\_rules("email", "email", "required");

$this->form\_validation->set\_rules("no\_hp", "No Handphone", "required|numeric");

$this->form\_validation->set\_rules("alamat", "Alamat", "required");

$this->form\_validation->set\_rules("password", "Password", "required");

$this->form\_validation->set\_rules("role\_id", "role id", "required");

$this->form\_validation->set\_rules("dapil\_id", "Dapil", "required");

$this->form\_validation->set\_rules("komisi\_id", "Komisi", "required");

$this->form\_validation->set\_rules("jabatan", "jabatan", "required");

if ($this->form\_validation->run()) {

$post = $this->input->post();

$image = $user['image'];

if (isset($\_FILES['image'])) {

$config['allowed\_types'] = 'gif|jpg|jpeg|png';

$config['max\_size'] = '3048';

$config['upload\_path'] = './assets/img/profile/';

$this->load->library('upload', $config);

if ($this->upload->do\_upload('image')) {

$image = $this->upload->data('file\_name');

/\*\*

\* Helper CI

\* Create thumb image with

\* \_img\_create\_thumbs({nama file image}, {Folder file gambar});

\*/

\_img\_create\_thumbs($image, "profile");

}

}

$tblUser = initTable("tbl\_user", "user");

$tblUser['lastkey'] = $tblUser['field'][$tblUser['key']];

$data = [];

foreach ($tblUser['field'] as $key => $val) {

if (array\_key\_exists($key, $this->input->post())) {

$data[$key] = $key == "password" ? password\_hash($post['password'], PASSWORD\_DEFAULT) : $this->input->post($key);

}

}

$data['image'] = $image;

$status = false;

$this->db->update($tblUser['name'], $data, ["id\_user" => $id]);

$respon = hasilCUD("Data User Di Update");

if ($respon->status)

$status = true;

$tbl = initTable("web\_komisi\_user", "kom\_u");

$data = [];

foreach ($tbl['field'] as $key => $val) {

if (array\_key\_exists($key, $this->input->post())) {

$data[$key] = $this->input->post($key);

}

}

$this->db->update("web\_komisi\_user", $data, ["user\_id" => $id]);

$respon = hasilCUD("Data User Di Update");

$status = $respon->status ? true : ($status ? true : false);

$type = "danger";

$message = "Tidak ada yang berubah";

$status ? [$type = "success", $message = "Data User Di Update"] : '';

$this->session->set\_flashdata('message', '<div class="alert alert-' . $type . '" role="alert">' . $message . '!</div>');

} else {

$this->session->set\_flashdata('message', '<div class="alert alert-danger" role="alert">Gagal..!</div>');

}

} else {

$this->session->set\_flashdata("message", "<div class='alert alert-danger' role='alert'>User Tidak Di temukan</div>");

}

redirect("admin/komisi/user");

Untuk mengolah aspirasi, menampilkan dan memberikan tanggapan dapat dilihat pada file Aspirasi.php dibawah ini

Aspirasi.php

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Aspirasi extends Admin\_Controller

{

public function \_\_construct()

{

parent::\_\_construct();

$this->post = $this->input->post();

$this->get = $this->input->get();

}

public function index()

{

$this->data['page']['title'] = 'Daftar Aspirasi Yang Masuk';

$this->data['page']['description'] = 'Silahkan edit menu Atau Tambahkan Komisi.!';

// $this->data['page']['before'] = ['url' => base\_url('admin/admin/role'), "title" => "Role Access"];

$this->data['page']['submenu'] = 'Menu Komisi';

$this->db->select("web\_komisi\_user.\*,web\_komisi.name");

$this->db->from("web\_komisi\_user");

$this->db->join("web\_komisi", "web\_komisi.id\_komisi=web\_komisi\_user.komisi\_id");

$this->db->where("web\_komisi\_user.user\_id", $this->data['user']['id\_user']);

$eks = $this->db->get()->row\_array();

$this->data['komisi'] = $eks;

$this->template->load('admin', 'aspirasi/komisi', $this->data);

}

public function delete($id)

{

$komisi = $this->komisi\_m;

$komisi->setPrimaryKey($id);

$komisi->deleted();

hasilCUD("Sukses Menghapus Komisi");

redirect(base\_url('admin/komisi'));

}

public function edit($id)

{

$komisi = $this->komisi\_m;

$this->post[$komisi->getKeyName()] = $id;

$komisi->setFieldTable($this->post);

$komisi->update();

hasilCUD("Berhasil Update Komisi");

redirect(base\_url('admin/komisi'));

}

public function reply($id)

{

}

public function user()

{

$this->data['page']['title'] = 'Mengolah data User Komisi';

$this->data['page']['description'] = 'Silahkan edit User Komisi dan menonaktifkan atau mengaktifkan.!';

$this->data['page']['before'] = ['url' => base\_url('admin/komisi'), "title" => "Komisi"];

$this->data['page']['submenu'] = 'Daftar User untuk komisi';

$user = $this->tbl;

$user->setTable("tbl\_user", "user");

$field = [

"role\_id" => "2"

];

$user->setField($field);

$eks = $user->getWhere("role\_id", [

"join" => [

["tbl\_user\_file", 'tbl\_user']

],

"select" => "tbl\_user\_file.image",

"start" => 0

]);

$komisi = $this->tbl;

$komisi->setTable("web\_komisi");

$kms = $komisi->getWhere();

if ($eks->status) {

$this->data['all\_komisi'] = $kms->data;

$this->data['komisi\_user'] = $eks->data;

$this->template->load('admin', 'komisi/user', $this->data);

} else

var\_dump($eks);

}

public function userSet()

{

$user = new $this->tbl;

$user->setTable("tbl\_user", "user");

// $data = [

// "id\_user" => "user\_002",

// "is\_active" => "1"

// ];

// $user->setField($data);

$user->setField($this->post);

$user->selectField("is\_active");

$eks = $user->update();

echo json\_encode($eks);

}

function setkomisi()

{

$komisi = new $this->tbl; //load new model

$komisi->setTable("web\_komisi\_user", "k\_u\_a"); // load table dan key exp: k\_u\_a\_001

$data = [

"user\_id" => $this->post['user\_id'],

"komisi\_id" => $this->post['komisi\_id'],

'status' => 1

];

$tbl = "web\_komisi\_user";

$response = (object) ['status' => false, 'message' => 'kesalahan Sistem'];

// $eks = $this->db->get\_where($tbl, ["user\_id" => $this->post['user\_id']]);

$eks = $this->db->get\_where($tbl, ["user\_id" => $data['user\_id']]);

$eks ? [

$eks->num\_rows() < 1 ? [

$data[$komisi->getKey()] = $komisi->getLastKey(),

$this->db->insert($tbl, $data),

$response = hasilCUD("Sukses Menambah User Ke komisi.!")

] : [

$this->db->update($tbl, $data, ["user\_id" => $data['user\_id']]),

$response = hasilCUD("Sukses Mengubah komisi User.!")

]

] : [];

echo json\_encode($response);

}

function userDelete($id)

{

$user = $this->tbl;

$user->setTable("tbl\_user", "user");

$user->setPrimaryKey($id);

$eks = $user->deleted();

echo json\_encode($eks);

}

}

Untuk file pendukun :

Komisi.php

<!DOCTYPE html>

<html lang="en">

<?php

$this->load->view($thema\_load . 'element/template/head\_meta.php');

?>

<body class="">

<!-- sidebar -->

<?php $this->load->view($thema\_load . 'element/template/sidebar.php'); ?>

<!-- end sidebara -->

<?php

$this->load->view($thema\_load . 'element/template/navbar.php');

?>

<!-- isi content -->

<div class="card">

<div class="card-header card-header-primary card-header-icon">

<div class="card-icon">

<i class="material-icons">assignment</i>

</div>

<h4 class="card-title">Daftar Aspirasi Masyarakat Pada Komisi <?= $komisi['name'] ?></h4>

</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-hover data-table">

<thead class="load">

<tr>

<th scope="col">#</th>

<th scope="col">User</th>

<th scope="col">Message</th>

<th scope="col">Status</th>

<th scope="col">Action</th>

</tr>

</thead>

<tbody class="daftar-aspirasi">

</tbody>

</table>

</div>

</div>

</div>

<!-- end isi -->

<?php

$this->load->view($thema\_load . 'element/template/footer.php');

?>

<?php

$this->load->view($thema\_load . 'element/template/fixed-setting.php');

?>

<div class="modal fade" id="lihat" tabindex="-1" role="dialog" aria-labelledby="#modalLabel" aria-hidden="true">

<div class="modal-dialog modal-lg" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="modalLabel">Pesan Aspirasi</b></h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body content-aspirasi">

<div class="row">

<div class="col-md-12">

<div class="card card-testimonial border">

<div class="icon">

<i class="material-icons">format\_quote</i>

</div>

<div class="card-body">

<h5 class="card-description">

</h5>

</div>

<div class="card-footer clear">

<h4 class="card-title">Alec Thompson</h4>

<div class="card-avatar">

<a href="#pablo">

<img class="img image" src="">

</a>

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="post-complaint mt-5">

<div class="complaint-meta d-block w-100">

<span class="meta-info tanggapi" data-url='4'>Tanggapi</span>

<span class="meta-info getInfo" id="tanggapan" data-role="2" data-url="">Tanggapan</span>

<span class="meta-info getInfo" id="komentar" data-role="3" data-url="">Komentar</span>

</div>

</div>

<div class="complaint-content">

<!-- Contenedor Principal -->

<div class="container">

<div class="row">

<!-- <h1>Comentarios <a href="http://creaticode.com">creaticode.com</a></h1> -->

<ul id="comments-list" class="comments-list w-100">

</ul>

</div>

</div>

<div class="default my-3 alert alert-danger" role='alert'>

Mohon maaf jika ada kesalahan dalam sistem kami. </br> silahakan anda melaporkannya agar kami bisa memperbaikinya

</div>

</div>

</div>

</div>

<div class="modal-footer">

</div>

</div>

</div>

</div>

<script>

var dftAspirasi = $("tbody.daftar-aspirasi");

const user = <?= json\_encode($user) ?>;

var urlLoadAspirasi = baseUrl + "api/aspirasi?id\_komisi=<?= $komisi['komisi\_id'] ?>"

var tr = null;

</script>

<script src="<?= $thema\_folder ?>layout/aspirasi/komisi.js"></script>

<script src="<?= $thema\_folder ?>assets/js/plugins/jquery.validate.min.js"></script>

</body>

</html>

Komisi.js

var \_aspirasi = null;

var aspirasi\_all = null;

var aspirasi\_selected = null;

var komentar\_all = null;

var komentar\_selected = null;

var \_list = null;

dftAspirasi.on("click", ".mdl#lihat", function() {

let \_parent = $(this).parents("tr");

let message = \_parent.children("td[name=message]").text();

let name = \_parent.children("td[name=username]").text();

let status = \_parent.children("td[name=status]").data("status");

let image = $(this).data("image");

let url = baseUrl + "api/aspirasi?id\_aspirasi=" + $(this).data("id");

let komisi\_id = $(this).data("komisi");

aspirasi\_selected = aspirasi\_all.find(obj => {

return obj.id\_aspirasi === $(this).data("id");

});

aspirasi\_selected = {

id\_aspirasi: $(this).data("id"),

status: \_parent.children("td[name=status]").data("status"),

message: \_parent.children("td[name=message]").text(),

image: $(this).data("image"),

komisi\_id: $(this).data("komisi")

};

$(".meta-info#komentar").data(

"url",

baseUrl + "api/komentar?aspirasi\_id=" + $(this).data("id")

);

$(".meta-info#tanggapan").data(

"url",

baseUrl + "api/komentar?aspirasi\_id=" + $(this).data("id")

);

if (status != 1 && status != 2) {

$.ajax({

url: url,

type: "PUT",

data: {

status: "2"

},

dataType: "json"

});

loadAspirasi();

}

$(".content-aspirasi")

.find(".card-description")

.text(message);

$(".content-aspirasi")

.find(".card-title")

.text(name);

$(".content-aspirasi")

.find(".image")

.attr("src", image);

htmlTanggapi();

empty();

});

$().ready(function() {

loadAspirasi();

});

function htmlTanggapi(myDiv = $("#comments-list")) {

$("#comments-list")

.find(".submit.tanggapi")

.remove();

let html = `

<form action='' method='post' class='submit tanggapi'>

<div class="form-group">

<label for="textKomentar">Tanggapan Anda</label>

<textarea class="form-control" id="textKomentar" name='komentar' rows="3"></textarea>

</div>

<div class="form-group">

<input type='submit' value='Send'/>

</form>

`;

myDiv.append(html);

$("#comments-list .submit.tanggapi").validate({

rules: {

komentar: {

required: true,

minlength: 12

},

action: "required"

},

messages: {

komentar: {

required: "Please enter some data",

minlength: "Your data must be at least 12 characters"

},

action: "Please provide some data"

},

submitHandler: function(form) {

async function putAspirasi(url = "", data = {}) {

// Default options are marked with \*

const response = await fetch(url, {

method: "PUT", // \*GET, POST, PUT, DELETE, etc.

mode: "cors", // no-cors, \*cors, same-origin

cache: "no-cache", // \*default, no-cache, reload, force-cache, only-if-cached

credentials: "same-origin", // include, \*same-origin, omit

headers: {

"Content-Type": "application/json"

// 'Content-Type': 'application/x-www-form-urlencoded',

},

redirect: "follow", // manual, \*follow, error

referrerPolicy: "no-referrer", // no-referrer, \*client

body: JSON.stringify(data) // body data type must match "Content-Type" header

});

return await response.json(); // parses JSON response into native JavaScript objects

}

if (aspirasi\_selected.status != 1 && komentar\_selected == null) {

putAspirasi(

baseUrl + "api/aspirasi?id\_aspirasi=" + aspirasi\_selected.id\_aspirasi,

{ status: 1 }

).then(data => {

loadAspirasi();

console.log(data); // JSON data parsed by `response.json()` call

});

}

let data = $(form).serializeArray();

data.push(

{

name: "aspirasi\_id",

value: aspirasi\_selected.id\_aspirasi

},

{

name: "user\_id",

value: user.id\_user

},

{

name: "parent",

value: komentar\_selected == null ? "0" : komentar\_selected.id\_komentar

}

);

let status,

message = null;

$.post(baseUrl + "api/komentar", data)

.done(res => {

if (res.status) {

loadAspirasi();

}

status = res.status;

message = res.message;

})

.fail(function(err) {

message = "Tanggapan Erros " + err;

})

.always(function() {

let type = status ? "success" : "warning";

let html = `

<div class="alert alert-${type}">

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<i class="material-icons">close</i>

</button>

<span>${message}</span>

</div>

`;

$("textarea[name=komentar]").val("");

$("#comments-list").prepend(html);

});

return false;

}

});

}

$(".tanggapi").click(function(e) {

$(".meta-info").removeClass("is-active");

$("#comments-list").html("");

komentar\_selected == null;

komentar\_all == null;

$(this).addClass("is-active");

e.preventDefault();

htmlTanggapi();

});

function getTanggal(dateInt) {

let d = new Date(parseInt(dateInt, 10) \* 1000);

let days = [

"Sunday", //senin

"Monday", //selasa

"Tuesday", //rabu

"Wednesday", //kamis

"Thursday", //jumat

"Friday", //sabtu

"Saturday" //minggu

];

let months = [

"January",

"February",

"March",

"April",

"May",

"June",

"July",

"August",

"September",

"October",

"November",

"December"

];

return days[d.getDay()] + "/" + months[d.getMonth()] + "/" + d.getFullYear();

}

$(".getInfo").click(function(e) {

$(".meta-info").removeClass("is-active");

e.preventDefault();

let url = $(this).data("url");

$.ajax({

url: url,

type: "GET",

data: {},

dataType: "json",

beforeSend: () => {

$(this).addClass("on-process");

$(".complaint-content .content").html("Wait.....!");

},

success: dt => {

if (dt.status) {

var list = "";

reply = "";

komentar\_all = dt.data;

dt.data.forEach(row => {

if ($(this).data("role") == row.role\_id && row.parent == 0) {

let date\_created = getTanggal(row.date\_created);

list += `<li id="tanggapan-root" data-id="${row.id\_komentar}">

<div class="comment-main-level">

<div class="comment-avatar"><img src="${row.image}" alt=""></div>

<div class="comment-box">

<div class="comment-head">

<h6 class="comment-name by-author"><a href="#">${row.username}</a></h6>

<span>${date\_created}</span>

<i class="material-icons kmt-delete mt-0">delete\_forever</i>

<i class="fa fa-reply kmt-reply" data-id="${row.id\_komentar}"></i>

</div>

<div class="comment-content">

${row.komentar}

</div>

</div>

</div>

<ul class="comments-list reply-list">

`;

let reply = listReplies(row.id\_komentar, dt.data);

list += reply;

}

list += "</ul></li>";

// list += reply;

});

$("#comments-list").html(list);

} else {

$("#comments-list").html("Tidak Ada Komentar");

}

$(this).addClass("is-active");

// status = dt.status;

// message = dt.message;

},

complete: function() {

$(".meta-info").removeClass("on-process");

}

});

});

let ReplyKomentar = $(".comments-list").on("click", ".kmt-reply", function() {

// tentukan id komentar target

let elRoot = $(this).parents("#tanggapan-root");

let id\_komentar = elRoot.data("idkomentar");

console.log(id\_komentar);

let parent = $(this).parents(".comment-main-level");

komentar\_selected = komentar\_all.find(obj => {

return obj.id\_komentar === id\_komentar;

});

let htmlRefly = `

`;

htmlTanggapi(parent);

});

$("#formInput").on("submit", function(e) {

e.preventDefault();

var myForm = $(this);

let url = myForm.attr("action");

var data = $(this).serializeArray(); //serialize form inputs and pass them to php

$.ajax({

url: url,

type: "PUT",

data: data,

dataType: "json",

beforeSend: () => {

myForm.after("<div class='mdlProgresBarr'></div>");

$(".fullpage").addClass("tampil");

},

success: dt => {

let userId = $('input[name="user\_id"]').val();

let nameKomisi = $("select[name='komisi\_id'] option:selected").text();

if (dt.status) {

\_aspirasi

.children("td[name=komisi\_id]")

.text($("#komisi\_id option:selected").text());

\_aspirasi

.children("td[name=message]")

.text($("textarea[name=message]").val());

}

status = dt.status;

message = dt.message;

},

complete: function() {

$("head").append("<style>.mdlProgresBarr::before{width:100%;}</style>");

$(".mdlProgresBarr").remove();

$(".modal").modal("hide");

let css = status ? "success" : "danger";

let notif = `<div class="alert alert-${css}">

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<i class="material-icons">close</i>

</button>

<span>${message}</span>

</div>`;

$(".fullpage").hide("slow", function() {

$(".fullpage").remove();

});

$(".notif#ajax").html(notif);

$(".main-panel").animate(

{

scrollTop: $(".notif#ajax").offset().top

},

"slow"

);

}

});

});

loadFileJs("layout/aspirasi/elemen/card\_komentar.js");

function listReplies(parent, list) {

var reply = ``;

list.forEach(element => {

if (element.parent == parent) {

reply += cardKomentar(element);

reply += listReplies(element.id\_komentar, list);

}

});

return reply;

}

function loadAspirasi() {

$.ajax({

url: urlLoadAspirasi,

type: "get",

dataType: "json",

beforeSend: () => {

$("thead.load").after("<div class='mdlProgresBarr'></div>");

},

success: respon => {

if (respon.status) {

let Urut = 1;

let tr = "";

aspirasi\_all = respon.data;

respon.data.forEach(e => {

let status =

e.status == 1

? ["success", "Ditanggapi"]

: e.status == 2

? ["warning", "Belum Di tanggapi"]

: ["danger", "Belum di Baca"];

tr += `

<tr id="${e.id\_aspirasi}">

<th scope="row">${Urut}</th>

<td name='username'>${e.username}</td>

<td name='message'>${e.message}</td>

<td name='status' data-status="${e.status}"><span class="badge badge-${status[0]}">${status[1]}</span></td>

<td>

<a href="" data-image="${e.image}" class="mdl badge badge-danger" id="lihat" data-id="${e.id\_aspirasi}" data-toggle="modal" data-url="${baseUrl}api/aspirasi?id\_aspirasi=${e.id\_aspirasi}" data-target=".modal#lihat">Lihat</a>

</td>

</tr>

`;

Urut++;

});

dftAspirasi.html(tr);

} else {

dftAspirasi.html(`<tr><td col=5>${respon.message}</td></tr>`);

}

},

error: e => {

console.log(e);

},

complete: () => {

$(".mdlProgresBarr").remove();

}

});

}

function empty() {

$("#comments-list").html("");

}

1. **LOGIN ANDROID**

public class LoginActivity extends AppCompatActivity {  
 EditText editTextEmail, editTextPassword;  
 ProgressBar progressBar;  
 ProgressDialog progressDialog;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (SharedPrefManager.getInstance(this).isLoggedIn()) {  
 finish();  
 startActivity(new Intent(LoginActivity.this, MainActivity.class));  
  
 }  
 setContentView(R.layout.activity\_login);  
 progressDialog = new ProgressDialog(this);  
 progressDialog.setMessage("Loading..."); // Setting Message  
 progressDialog.setTitle("Memeriksa data"); // Setting Title  
 progressDialog.setProgressStyle(ProgressDialog.STYLE\_SPINNER); // Progress Dialog Style Spinner  
 progressDialog.setCancelable(false);  
  
  
  
  
 progressBar = (ProgressBar) findViewById(R.id.progressBar);  
 editTextEmail = (EditText) findViewById(R.id.textInputEmail);  
 editTextPassword = (EditText) findViewById(R.id.textInputPassword);  
  
  
 //if user presses on login  
 //calling the method login  
 findViewById(R.id.btnlogin).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 userLogin();  
 }  
 });  
  
 //if user presses on not registered  
 findViewById(R.id.textViewRegister).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 //open register screen  
 startActivity(new Intent(getApplicationContext(), RegisterActivity.class));  
 }  
 });  
 }  
  
 private void userLogin() {  
 //first getting the values  
 final String email = editTextEmail.getText().toString();  
 final String password = editTextPassword.getText().toString();  
  
 //validating inputs  
 if (TextUtils.isEmpty(email)) {  
 editTextEmail.setError("Please enter your username");  
 editTextEmail.requestFocus();  
 return;  
 }  
  
 if (TextUtils.isEmpty(password)) {  
 editTextPassword.setError("Please enter your password");  
 editTextPassword.requestFocus();  
 return;  
 }  
 progressDialog.show(); // Display Progress Dialog  
  
 //if everything is fine  
 StringRequest stringRequest = new StringRequest(Request.Method.POST, URLs.URL\_LOGIN,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 progressBar.setVisibility(View.GONE);  
  
 try {  
 //converting response to json object  
 JSONObject obj = new JSONObject(response);  
 Boolean status = obj.getBoolean("status");  
 //if no error in response  
 if (status==true) {  
 Toast.makeText(getApplicationContext(), "sukses : "+obj.getString("message"), Toast.LENGTH\_SHORT).show();  
  
 //getting the user from the response  
 JSONObject userJson = obj.getJSONObject("data");  
  
 //creating a new user object  
 User user = new User(  
 userJson.getString("id\_user"),  
 userJson.getString("name"),  
 userJson.getString("email"),  
 userJson.getString("alamat"),  
 userJson.getString("no\_hp"),  
 userJson.getString("role\_id"),  
 userJson.getString("is\_active"),  
 userJson.getString("password"),  
 userJson.getString("image\_sm"),  
 userJson.getString("image\_lg"),  
 userJson.getString("tgl\_lahir"),  
 userJson.getString("tentang\_saya")  
  
 );  
  
 //storing the user in shared preferences  
 SharedPrefManager.getInstance(getApplicationContext()).userLogin(user);  
 finish();  
 //starting the profile activity  
 startActivity(new Intent(LoginActivity.this, MainActivity.class));  
 } else {  
 progressDialog.dismiss();  
 Toast.makeText(getApplicationContext(), "Error :" +obj.getString("message"), Toast.LENGTH\_SHORT).show();  
 }  
 } catch (JSONException e) {  
 e.printStackTrace();  
 Toast.makeText(getApplicationContext(), "Error :" +e.toString(), Toast.LENGTH\_SHORT).show();  
 progressDialog.dismiss();  
  
 }  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.makeText(getApplicationContext(),"Error : " + error.toString(), Toast.LENGTH\_SHORT).show();  
 progressDialog.dismiss();  
  
 }  
 }) {  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String, String> params = new HashMap<>();  
 params.put("email", email);  
 params.put("password", password);  
 return params;  
 }  
 };  
  
 VolleySingleton.getInstance(this).addToRequestQueue(stringRequest);  
 }  
}

REGISTER ANDROID

public class RegisterActivity extends AppCompatActivity {  
  
 EditText editTextName, editTextEmail, editTextPassword,editTextTglLahir,editTextNoHp,editTextAlamat;  
 ProgressBar progressBar;  
 ProgressDialog progressDialog;  
 Calendar myCalendar;  
 DatePickerDialog.OnDateSetListener date;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_register);  
 progressDialog = new ProgressDialog(this);  
 progressDialog.setMessage("Loading..."); // Setting Message  
 progressDialog.setTitle("Register"); // Setting Title  
 progressDialog.setProgressStyle(ProgressDialog.STYLE\_SPINNER); // Progress Dialog Style Spinner  
 progressDialog.setCancelable(false);  
  
 progressBar = (ProgressBar) findViewById(R.id.progressBar);  
 editTextName = (EditText) findViewById(R.id.txtName);  
 editTextName.requestFocus();  
 editTextEmail = (EditText) findViewById(R.id.txtEmail);  
 editTextPassword = (EditText) findViewById(R.id.txtPassword);  
 editTextTglLahir = (EditText) findViewById(R.id.txtTgllahir);  
 editTextNoHp =(EditText) findViewById(R.id.txtNohp);  
 editTextAlamat =(EditText) findViewById(R.id.txtAlamat);  
  
 //if the user is already logged in we will directly start the profile activity  
 if (SharedPrefManager.getInstance(this).isLoggedIn()) {  
 finish();  
 startActivity(new Intent(RegisterActivity.this, MainActivity.class));  
 return;  
 }  
 findViewById(R.id.btnRegister).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 //if user pressed on button register  
 //here we will register the user to server  
 registerUser();  
 }  
 });  
 findViewById(R.id.textViewLogin).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 //if user pressed on login  
 //we will open the login screen  
 finish();  
 startActivity(new Intent(RegisterActivity.this, LoginActivity.class));  
 }  
 });  
 myCalendar = Calendar.getInstance();  
 date = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int monthOfYear,  
 int dayOfMonth) {  
 // TODO Auto-generated method stub  
 myCalendar.set(Calendar.YEAR, year);  
 myCalendar.set(Calendar.MONTH, monthOfYear);  
 myCalendar.set(Calendar.DAY\_OF\_MONTH, dayOfMonth);  
 view.setSpinnersShown(true);  
 updateLabel();  
 }  
 };  
 editTextTglLahir.setOnClickListener(new View.OnClickListener() {  
  
 @Override  
 public void onClick(View v) {  
 // TODO Auto-generated method stub  
 new DatePickerDialog(RegisterActivity.this, date, myCalendar  
 .get(Calendar.YEAR), myCalendar.get(Calendar.MONTH),  
 myCalendar.get(Calendar.DAY\_OF\_MONTH)).show();  
 }  
 });  
 }  
 private void updateLabel() {  
 String myFormat = "dd-MM-YYYY";  
 SimpleDateFormat sdf = new SimpleDateFormat(myFormat, Locale.US);  
 editTextTglLahir.setText(sdf.format(myCalendar.getTime()));  
 }  
 private void registerUser() {  
 final String name = editTextName.getText().toString().trim();  
 final String email = editTextEmail.getText().toString().trim();  
 final String noHp = editTextNoHp.getText().toString().trim();  
 final String tglLahir = editTextTglLahir.getText().toString().trim();  
 final String password = editTextPassword.getText().toString().trim();  
 final String alamat = editTextAlamat.getText().toString().trim();  
  
 //first we will do the validations  
 if (TextUtils.isEmpty(name)) {  
 editTextName.setError("Masukkan Nama Lengkap anda");  
 editTextName.requestFocus();  
 return;  
 }  
 if (TextUtils.isEmpty(email)) {  
 editTextEmail.setError("Please enter your email");  
 editTextEmail.requestFocus();  
 return;  
 }  
 if (!android.util.Patterns.EMAIL\_ADDRESS.matcher(email).matches()) {  
 editTextEmail.setError("Enter a valid email");  
 editTextEmail.requestFocus();  
 return;  
 }  
 if (TextUtils.isEmpty(noHp)) {  
 editTextNoHp.setError("Masukkan No Hp yang benar");  
 editTextNoHp.requestFocus();  
 return;  
 }  
 if (TextUtils.isEmpty(tglLahir)) {  
 editTextTglLahir.setError("Masukkan Tgl Lahir anda");  
 editTextTglLahir.requestFocus();  
 return;  
 }  
 if (TextUtils.isEmpty(password)) {  
 editTextPassword.setError("Enter a password");  
 editTextPassword.requestFocus();  
 return;  
 }  
 if (TextUtils.isEmpty(alamat)) {  
 editTextAlamat.setError("Enter a password");  
 editTextAlamat.requestFocus();  
 return;  
 }  
 progressDialog.show(); // Display Progress Dialog  
 progressBar.setVisibility(View.VISIBLE);  
 StringRequest stringRequest = new StringRequest(Request.Method.POST, URLs.URL\_REGISTER,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 try {  
 //converting response to json object  
 JSONObject obj = new JSONObject(response);  
 Boolean status = obj.getBoolean("status");  
 //if no error in response  
 if (status) {  
 Toast.makeText(getApplicationContext(), obj.getString("message")+"\n Silahkan verifikasi di email anda dan login kembali", Toast.LENGTH\_SHORT).show();  
 //getting the user from the response  
 //starting the profile activity  
 finish();  
 startActivity(new Intent(getApplicationContext(), LoginActivity.class));  
 } else {  
 Toast.makeText(getApplicationContext(),"Gagal : "+ obj.getString("message"), Toast.LENGTH\_SHORT).show();  
 progressBar.setVisibility(View.GONE);  
 progressDialog.dismiss();  
 }  
 } catch (JSONException e) {  
 e.printStackTrace();  
 progressBar.setVisibility(View.GONE);  
 progressDialog.dismiss();  
 Toast.makeText(getApplicationContext(),"Error JSON : "+ e.toString( ), Toast.LENGTH\_SHORT).show();  
  
 }  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 progressBar.setVisibility(View.GONE);  
 progressDialog.dismiss();  
  
 Toast.makeText(getApplicationContext(),"Error : " + error.toString() , Toast.LENGTH\_SHORT).show();  
 }  
 }) {  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String, String> params = new HashMap<>();  
 params.put("name", name);  
 params.put("email", email);  
 params.put("password", password);  
 params.put("alamat", alamat);  
 params.put("tgl\_lahir", tglLahir);  
 params.put("no\_hp", noHp);  
 return params;  
 }  
 };  
  
 VolleySingleton.getInstance(this).addToRequestQueue(stringRequest);  
  
 }  
}

1. **ASPIRASI**

public class AspirasiFragment extends Fragment {  
  
 public static Object Datalist;  
 private ProgressDialog pDialog ;// membuat loading  
 private User user;  
  
 public List<Aspirasi> listAspirasiJson = new ArrayList<>();  
  
  
 private Dialog dialog\_aspirasi;  
 private RecyclerView aList;  
 private LinearLayoutManager linearLayoutManager;  
  
 // Array LIst  
 private ArrayList<Kecamatan\_model> kecamatanList;  
 private ArrayList<String> arrayNameKec;  
  
 //Spinner  
 Spinner spinner,spinnerStatusAspirasi;  
  
  
 private List<Aspirasi> AspirasiList;  
 private RecyclerView.Adapter adapterAspirasi;  
  
 FloatingActionButton fbaspirasi;  
 private Button hapusasp;  
 private EditText inputaspirasi;  
 private Button kirimaspirasi, cancelaspirasi;  
 RelativeLayout textKosong;  
  
 public AspirasiFragment() {  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 View root = inflater.inflate(R.layout.fragment\_aspirasi, container, false);  
 user = SharedPrefManager.getInstance(getActivity()).getUser();  
  
  
 pDialog = new ProgressDialog(getActivity());  
 pDialog.setMessage("Loading...");  
 pDialog.setCancelable(false);  
  
 if (listAspirasiJson.size()<1){  
 getJsonArray();  
 getKecamatan();  
 }  
  
 /\*\*  
 \* inisialisasi Recyclecardview  
 \*/  
 AspirasiList = new ArrayList<>();  
 //untuk Aspirasi Kosong  
 textKosong =root.findViewById(R.id.textViewKosong);  
 //---  
 aList =(RecyclerView) root.findViewById(R.id.aspirasi\_recycler\_view);  
 aList.setHasFixedSize(true);  
 //Kita menggunakan LinearLayoutManager untuk list standar  
 //yang hanya berisi daftar item  
 //disusun dari atas ke bawah  
 linearLayoutManager = new LinearLayoutManager(getActivity());  
 aList.setLayoutManager(linearLayoutManager);  
 adapterAspirasi = new AspirasiAdapter(getContext(),AspirasiList);  
 aList.setAdapter(adapterAspirasi);  
 //---==========================================================  
  
  
  
 /\*\*  
 \* setting Spinnner Status Aspirasi  
 \*/  
 spinnerStatusAspirasi = root.findViewById(R.id.spOpt);  
 spinnerStatusAspirasi.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 getData();  
 }  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
 }  
 });  
  
 fbaspirasi = root.findViewById(R.id.fab1);  
 fbaspirasi.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 initCustomDialog();  
 }  
 });  
 return root;  
 }  
 @Override  
 public void onViewCreated(View view, Bundle savedInstanceState) {  
  
 }  
  
 /\*\*  
 \* inisialisasi json array aspirasi  
 \*/  
 public void getJsonArray(){  
 pDialog.show();  
 String url = URLs.URL\_ASPIRASI +"/"+user.getIdUser();  
 StringRequest stringRequest = new StringRequest(Request.Method.GET, url,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 try {  
 //converting response to json object  
 JSONObject obj = new JSONObject(response);  
 Boolean status = obj.getBoolean("status");  
 //if no error in response  
 if (status) {  
 JSONArray json = obj.getJSONArray("data");  
 for (int i=0;i<json.length();i++){  
 Aspirasi asp = new Aspirasi(json.getJSONObject(i));  
 listAspirasiJson.add(asp);  
 }  
 getData();  
 } else {  
 Toast.makeText(getActivity(), "Status Error :" +obj.getString("message"), Toast.LENGTH\_SHORT).show();  
 }  
 pDialog.dismiss();  
 } catch (JSONException e) {  
 e.printStackTrace();  
 Toast.makeText(getActivity(), "Json Error :" +e.toString(), Toast.LENGTH\_SHORT).show();  
 pDialog.dismiss();  
 }  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.makeText(getActivity(),"Volley Error : " + error.toString(), Toast.LENGTH\_SHORT).show();  
 pDialog.dismiss();  
 }  
 });  
 VolleySingleton.getInstance(getActivity()).addToRequestQueue(stringRequest);  
 }  
  
 private void getData() {  
 AspirasiList.clear();  
 textKosong.setVisibility(View.GONE);  
 aList.setVisibility(View.VISIBLE);  
  
  
 for (int i = 0; i < listAspirasiJson.size(); i++) {  
 String status = spinnerStatusAspirasi.getSelectedItemPosition()==0 ? "" : spinnerStatusAspirasi.getSelectedItemPosition()==2 ? "1" : "4";  
 Aspirasi aspirasi = listAspirasiJson.get(i);  
 if (status==""){  
 AspirasiList.add(aspirasi);  
 } else if(status.matches(aspirasi.getStatus())){  
 AspirasiList.add(aspirasi);  
 }  
 }  
 if (AspirasiList.size()<1){  
 textKosong.setVisibility(View.VISIBLE);  
 return;  
 }  
 adapterAspirasi.notifyDataSetChanged();  
  
 }  
  
  
 private String kecId;  
 private void initCustomDialog(){  
 dialog\_aspirasi = new Dialog(getActivity());  
 dialog\_aspirasi.requestWindowFeature(Window.FEATURE\_NO\_TITLE);  
 dialog\_aspirasi.setContentView(R.layout.inputaspirasi\_dialog);  
 dialog\_aspirasi.setCancelable(false);  
 WindowManager.LayoutParams lp = new WindowManager.LayoutParams();  
 lp.copyFrom(dialog\_aspirasi.getWindow().getAttributes());  
 lp.width = WindowManager.LayoutParams.MATCH\_PARENT;  
 dialog\_aspirasi.getWindow().setAttributes(lp);  
 dialog\_aspirasi.show();  
 spinner = dialog\_aspirasi.findViewById(R.id.spinnerKecamatan);  
 hapusasp = dialog\_aspirasi.findViewById(R.id.btn\_hapus\_aspirasi);  
 inputaspirasi = dialog\_aspirasi.findViewById(R.id.txt\_inputaspirasi);  
 kirimaspirasi = dialog\_aspirasi.findViewById(R.id.btn\_kirimaspirasi);  
 cancelaspirasi = dialog\_aspirasi.findViewById(R.id.btn\_cancelaspirasi);  
  
 List<String> listSpinner = new ArrayList<String>();  
 for (int i = 0; i < kecamatanList.size(); i++){  
 listSpinner.add(kecamatanList.get(i).getKecamatan().toString());  
 }  
 ArrayAdapter<String> adapters = new ArrayAdapter<String>(getActivity(),  
 android.R.layout.simple\_spinner\_item, listSpinner);  
 adapters.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);  
 spinner.setAdapter(adapters);  
  
 spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 //Setting the values to textviews for a selected item  
 Kecamatan\_model kecModel = (Kecamatan\_model) kecamatanList.get(position);  
 kecId = kecModel.getId\_kecamatan();  
 }  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
  
 }  
 });  
  
 kirimaspirasi.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 sendAsp();  
 }  
 });  
 cancelaspirasi.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 dialog\_aspirasi.dismiss();  
 }  
 });  
 }  
 private void sendAsp(){  
 pDialog.show();  
 StringRequest stringRequest = new StringRequest(Request.Method.POST, URLs.URL\_ASPIRASI,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response){  
 try {  
 JSONObject obj = new JSONObject(response);  
 Boolean status = obj.getBoolean("status");  
 //if no error in response  
 if (status) {  
 JSONArray jsonArray = obj.getJSONArray("data");  
 for (int i = 0; i < jsonArray.length(); i++) {  
 try {  
 Log.e("Aspirasi : ","Ditambah"+i);  
 JSONObject objAspirasi = jsonArray.getJSONObject(i);  
 Aspirasi aspirasi = new Aspirasi(objAspirasi);  
 listAspirasiJson.add(0,aspirasi);  
 }catch (JSONException e) {  
 pDialog.dismiss();  
 e.printStackTrace();  
 }  
 }  
  
 getData();  
  
 dialog\_aspirasi.dismiss();  
 pDialog.dismiss();  
 } else {  
 Toast.makeText(getActivity(),"Error : "+ obj.getString( "message"), Toast.LENGTH\_SHORT).show();  
 }  
 pDialog.dismiss();  
 }catch (JSONException e) {  
 e.printStackTrace();  
 pDialog.dismiss();  
 Toast.makeText(getActivity(),"Error : "+ e.toString( ), Toast.LENGTH\_SHORT).show();  
 }  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 pDialog.dismiss();  
 Toast.makeText(getActivity(),"Error : " + error.toString() , Toast.LENGTH\_SHORT).show();  
 }  
 }){  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String, String> params = new HashMap<>();  
 params.put("user\_id", user.getIdUser());  
 params.put("message", inputaspirasi.getText().toString());  
 params.put("kec\_id", kecId);  
 return params;  
 }  
 };  
 VolleySingleton.getInstance(getActivity()).addToRequestQueue(stringRequest);  
 }  
 private void getKecamatan(){  
 StringRequest stringRequest = new StringRequest(Request.Method.GET, URLs.URL\_ALLKEC,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
  
 try {  
 //converting response to json object  
 JSONObject obj = new JSONObject(response);  
 Boolean status = obj.getBoolean("status");  
 //if no error in response  
 if (status) {  
 kecamatanList = new ArrayList<Kecamatan\_model>();  
 JSONArray dataArray = obj.getJSONArray("data");  
 for (int i = 0; i < dataArray.length(); i++) {  
 JSONObject dataobj = dataArray.getJSONObject(i);  
 Kecamatan\_model kecamatanModel = new Kecamatan\_model(dataobj);  
 kecamatanList.add(kecamatanModel);  
 }  
  
  
 } else  
 Toast.makeText(getActivity(),obj.getString("message"),Toast.LENGTH\_SHORT).show();  
 } catch (JSONException e) {  
 Toast.makeText(getActivity(),e.toString(),Toast.LENGTH\_SHORT).show();  
 e.printStackTrace();  
 }  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.makeText(getActivity(), error.getMessage(), Toast.LENGTH\_SHORT).show();  
 }  
 });  
  
 // request queue  
 VolleySingleton.getInstance(getActivity()).addToRequestQueue(stringRequest);  
 }  
}

1. **PROFIL**

public class ProfilFragment extends Fragment {  
 EditText txtNama,txtTglLahir,txtAlamat,txtNoHp,txtAboutme,txtEmail;  
 TextView txtVName ;  
 ImageView imgProfile;  
 ProgressDialog progressDialog;  
 Bitmap bitmap;  
 private User user;  
 Context applicationContext = MainActivity.getContextOfApplication();  
 Calendar myCalendar;  
 DatePickerDialog.OnDateSetListener date;  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 progressDialog = new ProgressDialog(getActivity());  
 progressDialog.setMessage("Loading..."); // Setting Message  
 progressDialog.setTitle("Menyimpan Data"); // Setting Title  
 progressDialog.setProgressStyle(ProgressDialog.STYLE\_SPINNER); // Progress Dialog Style Spinner  
 progressDialog.setCancelable(false);  
  
 View view = inflater.inflate(R.layout.fragment\_profil, container, false);  
 user = SharedPrefManager.getInstance(getActivity()).getUser();  
 txtAlamat = view.findViewById(R.id.editTextAlamat);  
 txtAlamat.setText(user.getAlamat());  
  
 txtTglLahir = view.findViewById(R.id.editTextTglLahir);  
 txtTglLahir.setText(user.getTglLahir());  
  
 txtNama = view.findViewById(R.id.editTextNama);  
 txtNama.setText(user.getName());  
  
 txtAboutme= view.findViewById(R.id.editTextAbout);  
 txtAboutme.setText(user.getAbout());  
  
 txtNoHp= view.findViewById(R.id.editTextNoTelp);  
 txtNoHp.setText(user.getNoHp());  
  
 txtEmail = view.findViewById(R.id.editTextEmail);  
 txtEmail.setText(user.getEmail());  
  
 txtVName = view.findViewById(R.id.textViewName);  
 txtVName.setText(user.getName());  
  
 imgProfile = view.findViewById(R.id.imgViewprofile);  
 Picasso.with(getActivity())  
 .load(user.getImageSm())  
 .centerCrop()  
 .transform(new CircleTransform(50,0))  
 .fit()  
 .into(imgProfile);  
  
  
 view.findViewById(R.id.btnLogOut).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 getActivity().finish();  
 SharedPrefManager.getInstance(getActivity()).logout();  
 }  
 });  
 view.findViewById(R.id.btnSimpan).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 simpan();  
 }  
 });  
 view.findViewById(R.id.btnGantiPhoto).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 chooseFile();  
 }  
 });  
  
 myCalendar = Calendar.getInstance();  
 date = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int monthOfYear,  
 int dayOfMonth) {  
 // TODO Auto-generated method stub  
 myCalendar.set(Calendar.YEAR, year);  
 myCalendar.set(Calendar.MONTH, monthOfYear);  
 myCalendar.set(Calendar.DAY\_OF\_MONTH, dayOfMonth);  
 view.setSpinnersShown(true);  
 updateLabel();  
 }  
 };  
  
 txtTglLahir.setOnClickListener(new View.OnClickListener() {  
  
 @Override  
 public void onClick(View v) {  
 // TODO Auto-generated method stub  
 new DatePickerDialog(getActivity(), date, myCalendar  
 .get(Calendar.YEAR), myCalendar.get(Calendar.MONTH),  
 myCalendar.get(Calendar.DAY\_OF\_MONTH)).show();  
 }  
 });  
 return view;  
  
 }  
 private void updateLabel() {  
 String myFormat = "dd-MM-YYYY";  
 SimpleDateFormat sdf = new SimpleDateFormat(myFormat, Locale.US);  
 txtTglLahir.setText(sdf.format(myCalendar.getTime()));  
 }  
  
 private void simpan() {  
 progressDialog.show(); // Display Progress Dialog  
 StringRequest stringRequest = new StringRequest(Request.Method.PUT, URLs.URL\_UPDATEPROFILE + user.getIdUser(),  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 try {  
 //converting response to json object  
 JSONObject obj = new JSONObject(response);  
 Boolean status = obj.getBoolean("status");  
 //if no error in response  
 if (status==true) {  
 Toast.makeText(getActivity(), "sukses : "+obj.getString("message"), Toast.LENGTH\_SHORT).show();  
  
 //getting the user from the response  
 JSONObject userJson = obj.getJSONObject("data");  
 //creating a new user object  
 //creating a new user object  
 User user = new User(  
 userJson.getString("id\_user"),  
 userJson.getString("name"),  
 userJson.getString("email"),  
 userJson.getString("alamat"),  
 userJson.getString("no\_hp"),  
 userJson.getString("role\_id"),  
 userJson.getString("is\_active"),  
 userJson.getString("password"),  
 userJson.getString("image\_sm"),  
 userJson.getString("image\_lg"),  
 userJson.getString("tgl\_lahir"),  
 userJson.getString("tentang\_saya")  
  
 );  
 //storing the user in shared preferences  
 SharedPrefManager.getInstance(getActivity()).userLogin(user);  
//  
 } else {  
 Toast.makeText(getActivity(), "Error :" +obj.getString("message"), Toast.LENGTH\_SHORT).show();  
 }  
 progressDialog.dismiss();  
 } catch (JSONException e) {  
 e.printStackTrace();  
 Toast.makeText(getActivity(), "Error :" +e.toString(), Toast.LENGTH\_SHORT).show();  
 progressDialog.dismiss();  
 }  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.makeText(getActivity(),"Error : " + error.toString(), Toast.LENGTH\_SHORT).show();  
 progressDialog.dismiss();  
 }  
 }) {  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String, String> params = new HashMap<>();  
 params.put("name",txtNama.getText().toString());  
 params.put("alamat",txtAlamat.getText().toString());  
 params.put("tgl\_lahir",txtTglLahir.getText().toString());  
 params.put("no\_hp",txtNoHp.getText().toString());  
 params.put("tentang\_saya",txtAboutme.getText().toString());  
 return params;  
 }  
 };  
 VolleySingleton.getInstance(getActivity()).addToRequestQueue(stringRequest);  
 }  
 private void chooseFile(){  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.ACTION\_GET\_CONTENT);  
 startActivityForResult(Intent.createChooser(intent,"Select Picture"),1);  
 }  
 @Override  
 public void onActivityResult(int requestCode, int resultCode, Intent data){  
 super.onActivityResult(requestCode,resultCode,data);  
 if (requestCode==1 && resultCode==RESULT\_OK && data != null && data.getData() !=null){  
 Uri filePath =data.getData();  
 try {  
 bitmap = MediaStore.Images.Media.getBitmap(getActivity().getContentResolver(),filePath);  
 imgProfile.setImageBitmap(bitmap);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 uploadPicture(user.getIdUser(),getStringImage(bitmap));  
  
  
 }  
 }  
  
 private String getStringImage(Bitmap bitmap) {  
 ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();  
 bitmap.compress(Bitmap.CompressFormat.JPEG,50,byteArrayOutputStream);  
 byte[] imageByteArray = byteArrayOutputStream.toByteArray();  
 String encodeImage = Base64.encodeToString(imageByteArray,Base64.DEFAULT);  
 return encodeImage;  
 }  
  
 private void uploadPicture(final String idUser, final String foto) {  
 final ProgressDialog pDialog = new ProgressDialog(getActivity());  
 pDialog.setMessage("Upload Foto..");  
 pDialog.show();  
 Toast.makeText(getActivity(),idUser, Toast.LENGTH\_SHORT).show();  
 StringRequest stringRequest = new StringRequest(Request.Method.POST, URLs.URL\_UPLOAD\_GAMBAR,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response){  
 pDialog.dismiss();  
 try {  
 JSONObject obj = new JSONObject(response);  
 Boolean status = obj.getBoolean("status");  
 //if no error in response  
 if (status) {  
// FragmentTransaction ft = getFragmentManager().beginTransaction();  
// ft.detach(ProfilFragment.this).attach(ProfilFragment.this).commit();  
 JSONObject img = obj.getJSONObject("data");  
 SharedPrefManager.getInstance(getActivity()).setImg(img.getString("imageSm"),img.getString("imageLg"));  
  
 } else {  
 Toast.makeText(getActivity(),"Gagal : "+ obj.getString("message"), Toast.LENGTH\_SHORT).show();  
 }  
 }catch (JSONException e) {  
 e.printStackTrace();  
  
 Toast.makeText(getActivity(),"Error : "+ e.toString( ), Toast.LENGTH\_SHORT).show();  
  
 }  
 }  
 },  
 new Response.ErrorListener() {  
  
  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 pDialog.dismiss();  
 Toast.makeText(getActivity(),"Error : " + error.toString() , Toast.LENGTH\_SHORT).show();  
 }  
 }){  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String, String> params = new HashMap<>();  
 params.put("id\_user", idUser);  
 params.put("image", foto);  
 return params;  
 }  
 };  
 VolleySingleton.getInstance(getActivity()).addToRequestQueue(stringRequest);  
  
  
 }  
  
  
}