**ADMIN - PAGE CONTROLLER :**

**mds-trackmate/application/controllers/admin/Authentication.php - code**

<?php

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

class Authentication extends App\_Controller

{

public function \_\_construct()

{

// die('success');

parent::\_\_construct();

if ($this->app->is\_db\_upgrade\_required()) {

redirect(admin\_url());

}

load\_admin\_language();

$this->load->model('Authentication\_model');

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

$this->form\_validation->set\_message('required', \_l('form\_validation\_required'));

$this->form\_validation->set\_message('valid\_email', \_l('form\_validation\_valid\_email'));

$this->form\_validation->set\_message('matches', \_l('form\_validation\_matches'));

hooks()->do\_action('admin\_auth\_init');

}

public function index()

{

// die('admi');

$this->admin();

}

public function admin()

{

// die('admin');

if (is\_staff\_logged\_in()) {

redirect(admin\_url());

}

$this->form\_validation->set\_rules('password', \_l('admin\_auth\_login\_password'), 'required');

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

if (show\_recaptcha()) {

$this->form\_validation->set\_rules('g-recaptcha-response', 'Captcha', 'callback\_recaptcha');

}

if ($this->input->post()) {

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

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

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

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

$data = $this->Authentication\_model->login($email, $password, $remember, true);

if (is\_array($data) && isset($data['memberinactive'])) {

set\_alert('danger', \_l('admin\_auth\_inactive\_account'));

redirect(admin\_url('authentication'));

} elseif (is\_array($data) && isset($data['two\_factor\_auth'])) {

if ($data['user']->two\_factor\_auth\_enabled == 1) {

$this->Authentication\_model->set\_two\_factor\_auth\_code($data['user']->staffid);

$sent = send\_mail\_template('staff\_two\_factor\_auth\_key', $data['user']);

if (!$sent) {

set\_alert('danger', \_l('two\_factor\_auth\_failed\_to\_send\_code'));

redirect(admin\_url('authentication'));

} else {

set\_alert('success', \_l('two\_factor\_auth\_code\_sent\_successfully', $email));

redirect(admin\_url('authentication/two\_factor'));

}

} else {

set\_alert('success', \_l('enter\_two\_factor\_auth\_code\_from\_mobile'));

redirect(admin\_url('authentication/two\_factor/app'));

}

} elseif ($data == false) {

set\_alert('danger', \_l('admin\_auth\_invalid\_email\_or\_password'));

//redirect(admin\_url('authentication'));

}

$this->load->model('announcements\_model');

$this->announcements\_model->set\_announcements\_as\_read\_except\_last\_one(get\_staff\_user\_id(), true);

// is logged in

maybe\_redirect\_to\_previous\_url();

hooks()->do\_action('after\_staff\_login');

redirect(admin\_url());

}

}

$data['title'] = \_l('admin\_auth\_login\_heading');

$this->load->view('authentication/login\_admin', $data);

}

public function two\_factor($type = 'email')

{

$this->form\_validation->set\_rules('code', \_l('two\_factor\_authentication\_code'), 'required');

if ($this->input->post()) {

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

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

$code = trim($code);

if ($this->Authentication\_model->is\_two\_factor\_code\_valid($code) && $type = 'email') {

$user = $this->Authentication\_model->get\_user\_by\_two\_factor\_auth\_code($code);

$this->Authentication\_model->clear\_two\_factor\_auth\_code($user->staffid);

$this->Authentication\_model->two\_factor\_auth\_login($user);

$this->load->model('announcements\_model');

$this->announcements\_model->set\_announcements\_as\_read\_except\_last\_one(get\_staff\_user\_id(), true);

maybe\_redirect\_to\_previous\_url();

hooks()->do\_action('after\_staff\_login');

redirect(admin\_url());

} elseif ($this->Authentication\_model->is\_google\_two\_factor\_code\_valid($code) && $type = 'app') {

$user = get\_staff($this->session->userdata('tfa\_staffid'));

$this->Authentication\_model->two\_factor\_auth\_login($user);

$this->load->model('announcements\_model');

$this->announcements\_model->set\_announcements\_as\_read\_except\_last\_one(get\_staff\_user\_id(), true);

maybe\_redirect\_to\_previous\_url();

hooks()->do\_action('after\_staff\_login');

redirect(admin\_url());

} else {

set\_alert('danger', \_l('two\_factor\_code\_not\_valid'));

redirect(admin\_url('authentication/two\_factor/' . $type));

}

}

}

$this->load->view('authentication/set\_two\_factor\_auth\_code');

}

public function forgot\_password()

{

if (is\_staff\_logged\_in()) {

redirect(admin\_url());

}

$this->form\_validation->set\_rules('email', \_l('admin\_auth\_login\_email'), 'trim|required|valid\_email|callback\_email\_exists');

if ($this->input->post()) {

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

$success = $this->Authentication\_model->forgot\_password($this->input->post('email'), true);

if (is\_array($success) && isset($success['memberinactive'])) {

set\_alert('danger', \_l('inactive\_account'));

redirect(admin\_url('authentication/forgot\_password'));

} elseif ($success == true) {

set\_alert('success', \_l('check\_email\_for\_resetting\_password'));

redirect(admin\_url('authentication'));

} else {

set\_alert('danger', \_l('error\_setting\_new\_password\_key'));

redirect(admin\_url('authentication/forgot\_password'));

}

}

}

$this->load->view('authentication/forgot\_password');

}

public function reset\_password($staff, $userid, $new\_pass\_key)

{

if (!$this->Authentication\_model->can\_reset\_password($staff, $userid, $new\_pass\_key)) {

set\_alert('danger', \_l('password\_reset\_key\_expired'));

redirect(admin\_url('authentication'));

}

$this->form\_validation->set\_rules('password', \_l('admin\_auth\_reset\_password'), 'required');

$this->form\_validation->set\_rules('passwordr', \_l('admin\_auth\_reset\_password\_repeat'), 'required|matches[password]');

if ($this->input->post()) {

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

hooks()->do\_action('before\_user\_reset\_password', [

'staff' => $staff,

'userid' => $userid,

]);

$success = $this->Authentication\_model->reset\_password($staff, $userid, $new\_pass\_key, $this->input->post('passwordr', false));

if (is\_array($success) && $success['expired'] == true) {

set\_alert('danger', \_l('password\_reset\_key\_expired'));

} elseif ($success == true) {

hooks()->do\_action('after\_user\_reset\_password', [

'staff' => $staff,

'userid' => $userid,

]);

set\_alert('success', \_l('password\_reset\_message'));

} else {

set\_alert('danger', \_l('password\_reset\_message\_fail'));

}

redirect(admin\_url('authentication'));

}

}

$this->load->view('authentication/reset\_password');

}

public function set\_password($staff, $userid, $new\_pass\_key)

{

if (!$this->Authentication\_model->can\_set\_password($staff, $userid, $new\_pass\_key)) {

set\_alert('danger', \_l('password\_reset\_key\_expired'));

if ($staff == 1) {

redirect(admin\_url('authentication'));

} else {

redirect(site\_url('authentication'));

}

}

$this->form\_validation->set\_rules('password', \_l('admin\_auth\_set\_password'), 'required');

$this->form\_validation->set\_rules('passwordr', \_l('admin\_auth\_set\_password\_repeat'), 'required|matches[password]');

if ($this->input->post()) {

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

$success = $this->Authentication\_model->set\_password($staff, $userid, $new\_pass\_key, $this->input->post('passwordr', false));

if (is\_array($success) && $success['expired'] == true) {

set\_alert('danger', \_l('password\_reset\_key\_expired'));

} elseif ($success == true) {

set\_alert('success', \_l('password\_reset\_message'));

} else {

set\_alert('danger', \_l('password\_reset\_message\_fail'));

}

if ($staff == 1) {

redirect(admin\_url('authentication'));

} else {

redirect(site\_url());

}

}

}

$this->load->view('authentication/set\_password');

}

public function logout()

{

$this->Authentication\_model->logout();

hooks()->do\_action('after\_user\_logout');

redirect(admin\_url('authentication'));

}

public function email\_exists($email)

{

$total\_rows = total\_rows(db\_prefix() . 'staff', [

'email' => $email,

]);

if ($total\_rows == 0) {

$this->form\_validation->set\_message('email\_exists', \_l('auth\_reset\_pass\_email\_not\_found'));

return false;

}

return true;

}

public function recaptcha($str = '')

{

return do\_recaptcha\_validation($str);

}

public function get\_qr()

{

if (!is\_staff\_logged\_in()) {

ajax\_access\_denied();

}

$company\_name = preg\_replace('/:/', '-', get\_option('companyname'));

if ($company\_name == '') {

// Colons is not allowed in the issuer name

$company\_name = rtrim(preg\_replace('/^https?:\/\//', '', site\_url()), '/') . ' - CRM';

}

$data = $this->authentication\_model->get\_qr($company\_name);

$this->load->view('admin/includes/google\_two\_factor', $data);

}

}