Звіт

до лабораторної роботи № 3

Розробка програмного продукту

Менеджер керування публічними посиланнями

Виконав:  
студент групи КН-31/2  
Манжула Давид

**Структура рішення**

Для розробки продукту використав MVC-фреймворк CodeIgniter.

Структура проекта:

├───application – Тут вся логіка

│ ├───config - Конфігурації

│ ├───controllers – Контроллери MVC

│ ├───libraries - Бібліотеки

│ │ ├───SxGeo – Бібліотека визлачення міста та країни по IP

│ │ └───LiqPay – Бібліотека платіжної системи

│ ├───models – Моделі MVC

│ └───views – В’юшки MVC

│ ├───errors – Сторінки помилок

│ ├───pages – Динамічні сторінки

│ └───static – Статичні сторінки

├───assets – Різні ресурси для Фронта – стилі, шрифти, зображення та Javascript

│ ├───css

│ │ ├───profile

│ │ └───style

│ ├───fonts

│ ├───img

│ │ └───icons

│ └───js

│ └───script

└───system

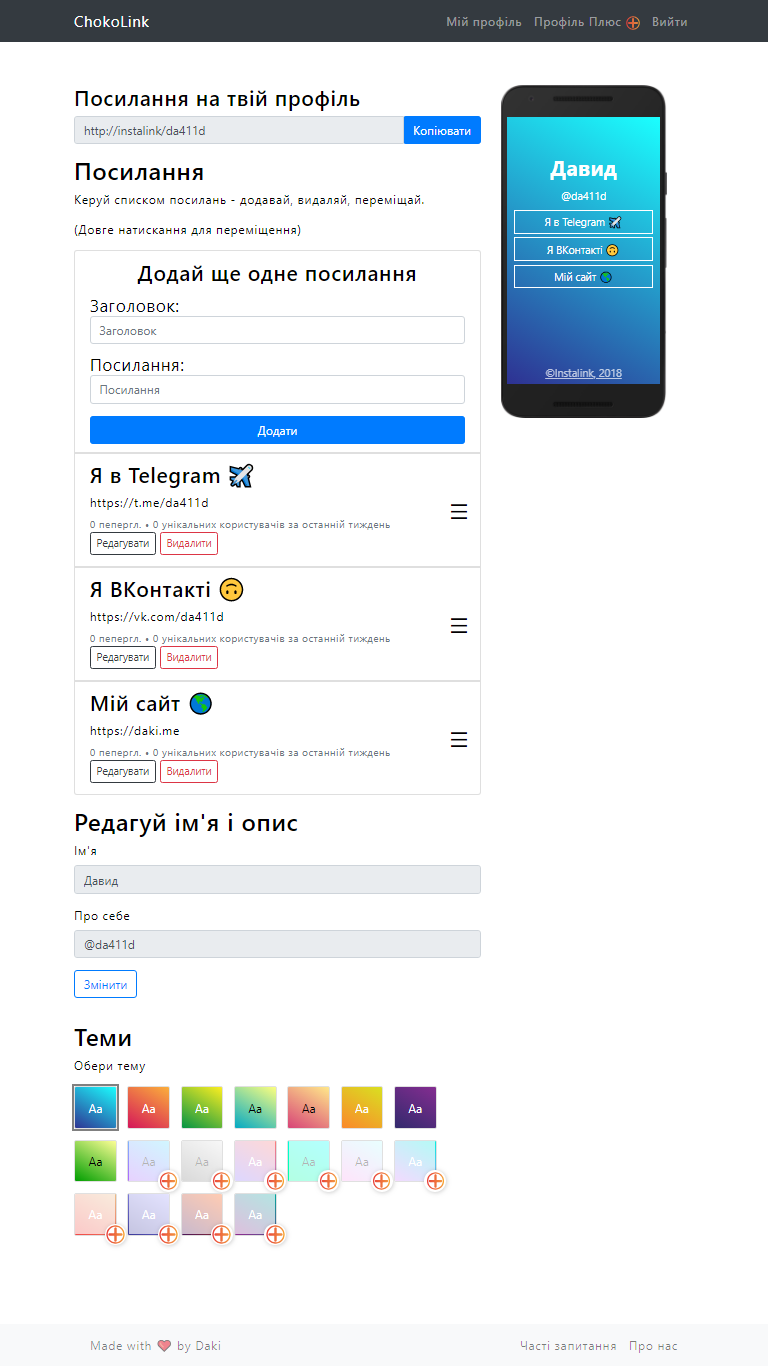
└─── Ядро фреймворка

**Головна форма застосунку:**

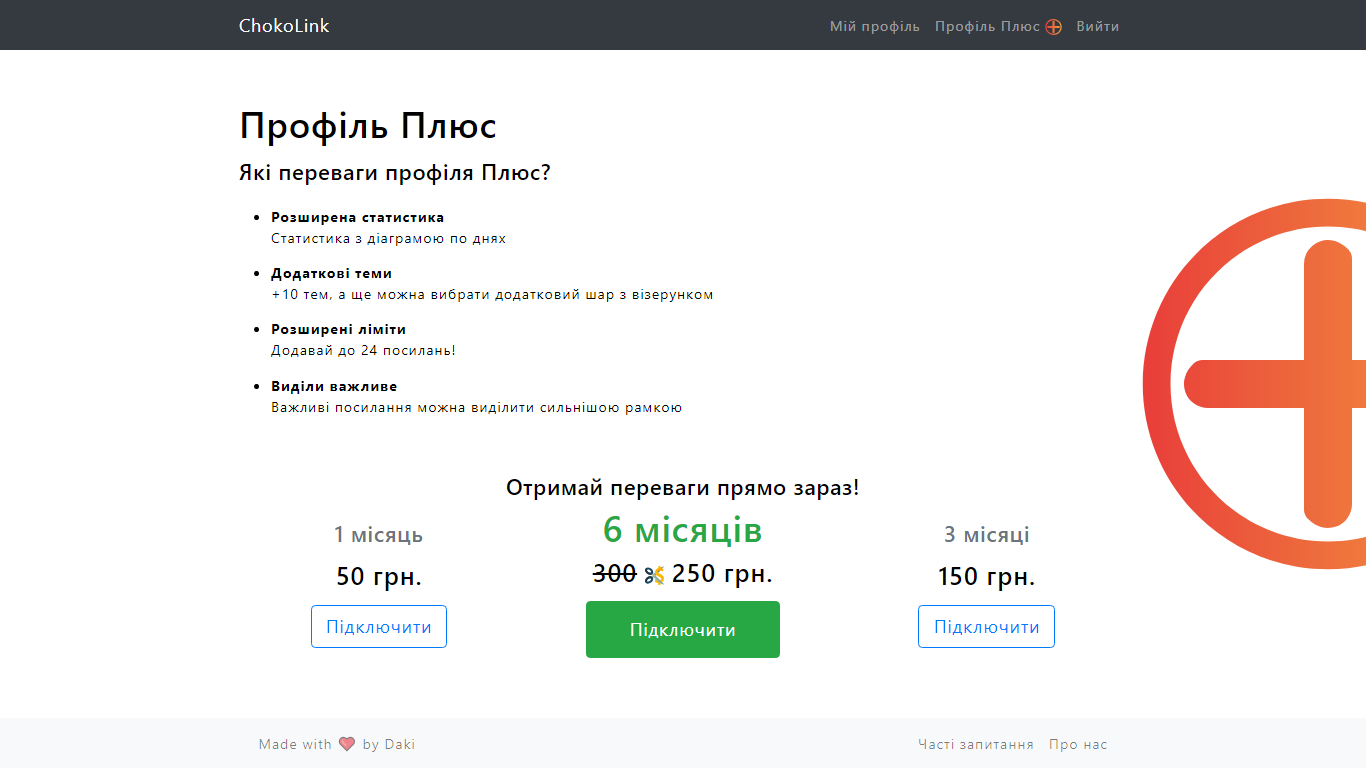
|  |
| --- |
| **Контроллер сторінки аккаунту** |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class Account extends CI\_Controller {  private $load\_status;  private $load\_message;  public function \_\_construct() {  parent::\_\_construct();  $this->load->model('profile\_model');  $this->load->model('core\_model');  $this->load->helper('url');  $this->load->helper(array('form', 'url'));  }  /\*  Генеружм шапочку  \*/  private function genHeader(&$data){  $user = $this->profile\_model->get\_user();    $data['username'] = $user["username"];  $data['header\_links'] = [];    $data['header\_links'][] = [  "title" => "Мій профіль",  "link" => "/account"  ];    if($user["plus"]){  $data['header\_links'][] = [  "title" => "Статистика",  "link" => "/account/statistics"  ];  }    $data['header\_links'][] = [  "title" => "Профіль Плюс <img class=\"i\" src=\"/assets/img/icons/plus.svg\">",  "link" => "/account/plus"  ];  $data['header\_links'][] = [  "title" => "Вийти",  "link" => "/account/logout"  ];    }    public function index($param = '' ){  if (!$this->profile\_model->get\_user()){  header("Location: /login");  return;  }  $data = [];    $this->genHeader($data);  $data["title"] = "Мій профіль";  $this->load->view('header', $data);  $this->load->view('pages/account', $data);  $this->load->view('footer', $data);  }  /\*  Логінимся  \*/  public function login(){  if($this->profile\_model->get\_user()){  header("Location: /account");  return;  }      $this->load->helper(array('form', 'url'));  $this->load->library('form\_validation');  $this->form\_validation->set\_rules('login', 'username', 'required');  $this->form\_validation->set\_rules('pass', 'password', 'required');  $username = $this->input->post('login');  $password = $this->input->post('pass');  if ( $username ) {  $data['username'] = $username;  } else {  $data['username'] = '';  }    if ($this->form\_validation->run() != FALSE) {  if ($this->profile\_model->login($username, $password)){  header("Location: /account");  } else {  $data['error\_mess'] = 'The username and password you entered did not match our records. Please try again.';  }  } else {  $data['error'] = $this->form\_validation->error\_array();  }  $data['title'] = 'Login';  $data["csrf"] = [  'name' => $this->security->get\_csrf\_token\_name(),  'hash' => $this->security->get\_csrf\_hash()  ];  $this->load->view('header', $data);  $this->load->view('pages/login', $data);  $this->load->view('footer', $data);  }  /\*  розлогінюємся  \*/  public function logout(){  $this->profile\_model->logout();  header("Location: /login");  }  /\*  Реєструємся  \*/  public function register(){  if($this->profile\_model->get\_user()){  header("Location: /account");  return;  }  $data = [];    $error\_mess = false;    $act = trim($this->security->xss\_clean($this->input->post('act')));    $name = trim($this->security->xss\_clean($this->input->post('name')));  $description = trim($this->security->xss\_clean($this->input->post('description')));  $login = trim($this->security->xss\_clean($this->input->post('login')));  $password\_id = trim($this->security->xss\_clean($this->input->post('rand')));  if(!strlen($password\_id))$password\_id = "1";  $password\_id = md5($password\_id\*2);  $password = trim($this->security->xss\_clean($this->input->post($password\_id)));    if($act == "register"){  try{  $this->profile\_model->registerUser([  "name" => $name,  "description" => $description,  "login" => $login,  "password" => $password  ]);  $this->profile\_model->login($login, $password);  if($this->profile\_model->get\_user()){  header("Location: /account");  return;  }      }catch(Exception $e){  $error\_mess = $e->getMessage();  }  }    $data["field\_salt"] = rand(111111, 99999999);  $data["field\_id"] = md5($data["field\_salt"]\*2);  $data["error\_mess"] = $error\_mess;    $data["name"] = $name;  $data["description"] = $description;  $data["login"] = $login;  $data["password"] = $password;    $data["csrf"] = [  'name' => $this->security->get\_csrf\_token\_name(),  'hash' => $this->security->get\_csrf\_hash()  ];  $this->load->view('header', $data);  $this->load->view('pages/register', $data);  $this->load->view('footer', $data);  }  /\*  Статистика  \*/  public function statistics($link\_id = false){  if(!$this->profile\_model->get\_user()){  header("Location: /login");  return;  }    $user = $this->profile\_model->get\_user();    if(!$user["plus"]){  header("Location: ./");  return;  }    $data = [];    $this->genHeader($data);  $data["title"] = "Статистика";  $data["links"] = $user["links"];    $this->load->view('header', $data);  $this->load->view('pages/statistics', $data);  $this->load->view('footer', $data);  }          /\*  Профіль плюс  \*/  public function plus($duration = false){  if(!$this->profile\_model->get\_user()){  header("Location: /login");  return;  }    $user = $this->profile\_model->get\_user();  $data = [];  $this->genHeader($data);    $data["title"] = "Профіль Плюс";  $data["plus"] = $user["plus"];  $data["plus\_expires"] = $user["plus\_expires"];    if($duration){  echo $this->core\_model->genPaylink($user["id"], $duration);  echo "<script>document.querySelector('#liqpay\_form').submit()</script>";    }else{  $data["features"] = $this->core\_model->getFeatures();  $data["payment\_duratoins"] = $this->core\_model->getPlusDurations();    $this->load->view('header', $data);  $this->load->view('pages/plus', $data);  $this->load->view('footer', $data);  }  }  /\*  Коллбек для liqpay  \*/  public function callback(){  $data = trim($this->security->xss\_clean($this->input->post('data')));  $signature = trim($this->security->xss\_clean($this->input->post('signature')));    $payment = $this->core\_model->checkPayment($data, $signature);    if($payment){  $this->profile\_model->updatePlus($payment["user\_id"], $payment["duration"]);  }  }  } |

**Результат користування**

Спершу користувач попадає на лендинг, де бачить заклик до реєстрації чи входу. Після реєстрації чи входу він потрапляє в особистий кабінет, де може скопіювати адресу на свою публічну сторінку, додати, видалити, редагувати та впорядкувати посилання, вибрати тему оформлення, змінити ім’я та опис сторінки.



Ще користувач може придбати розширений функціонал.



Після оплати він може переглядати статистику по днях та містах

