**1.Dashboard**

**-Controller:**

<?php

require\_once ("secure\_area.php");

class Home extends Secure\_area {

function \_\_construct()

{

parent::\_\_construct();

}

function index()

{

$data["total\_loans"] = $this->Loan->get\_total\_loans();

$data["total\_borrowers"] = $this->Customer->count\_all();

$data["my\_wallet"] = $this->My\_wallet->get\_total();

$this->load->view("home", $data);

}

function logout()

{

$this->Employee->logout();

}

}

?>

**-Model:**

function get\_total\_loans()

{

$this->db->select("SUM(loan\_amount) as total\_loans");

$this->db->from("loans");

$this->db->where("loan\_status", "pending");

$this->db->where("delete\_flag", "0");

$query = $this->db->get();

$res = $query->row();

return to\_currency($res->total\_loans, TRUE, 0);

}

function get\_total()

{

// credit

$user\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

$sql = "SELECT SUM(amount) AS amount FROM ".$this->db->dbprefix("wallets")." WHERE wallet\_type = 'credit' AND added\_by = ".$user\_id;

$query = $this->db->query($sql);

$credit = $query->row()->amount;

// debit

$sql = "SELECT SUM(amount) AS amount FROM ".$this->db->dbprefix("wallets")." WHERE wallet\_type = 'debit' AND added\_by = ".$user\_id;

$query = $this->db->query($sql);

$debit = $query->row()->amount;

// transfer to

$sql = "SELECT SUM(amount) AS amount FROM ".$this->db->dbprefix("wallets")." WHERE wallet\_type = 'transfer' AND added\_by = ".$user\_id." AND transfer\_to <> ".$user\_id;

$query = $this->db->query($sql);

$transfer\_to = $query->row()->amount;

// transfer to me

$sql = "SELECT SUM(amount) AS amount FROM ".$this->db->dbprefix("wallets")." WHERE wallet\_type = 'transfer' AND transfer\_to = ".$user\_id;

$query = $this->db->query($sql);

$transfer\_to\_me = $query->row()->amount;

return $debit - $credit - $transfer\_to + $transfer\_to\_me;

}

function count\_all()

{

$this->db->from('customers');

$this->db->where('deleted', 0);

return $this->db->count\_all\_results();

}

**-View:**

<?php

require\_once ("secure\_area.php");

class Home extends Secure\_area {

function \_\_construct()

{

parent::\_\_construct();

}

function index()

{

$data["total\_loans"] = $this->Loan->get\_total\_loans();

$data["total\_borrowers"] = $this->Customer->count\_all();

$data["my\_wallet"] = $this->My\_wallet->get\_total();

$this->load->view("home", $data);

}

function logout()

{

$this->Employee->logout();

}

}

?>

**2.Borrowers**

**-Controller:**

<?php

require\_once ("person\_controller.php");

class Customers extends Person\_controller {

function \_\_construct()

{

parent::\_\_construct('customers');

}

function index()

{

$config['base\_url'] = site\_url('/customers/index');

$config['total\_rows'] = $this->Customer->count\_all();

$config['per\_page'] = '20';

$config['uri\_segment'] = 3;

$this->pagination->initialize($config);

$res = $this->Employee->getLowerLevels();

$data['staffs'] = $res;

$data['controller\_name'] = strtolower(get\_class());

$data['form\_width'] = $this->get\_form\_width();

$data['manage\_table'] = get\_people\_manage\_table($this->Customer->get\_all($config['per\_page'], $this->uri->segment($config['uri\_segment'])), $this);

$this->load->view('people/manage', $data);

}

/\*

Returns customer table data rows. This will be called with AJAX.

\*/

function search()

{

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

$data\_rows = get\_people\_manage\_table\_data\_rows($this->Customer->search($search), $this);

echo $data\_rows;

}

/\*

Gives search suggestions based on what is being searched for

\*/

function suggest()

{

//$suggestions = $this->Customer->get\_search\_suggestions($this->input->post('q'), $this->input->post('limit'));

$suggestions = $this->Customer->get\_search\_suggestions($this->input->post('query'), 30);

//echo implode("\n", $suggestions);

$data = $tmp = array();

foreach ($suggestions as $suggestion):

$t = explode("|", $suggestion);

$tmp = array("value" => $t[1], "data" => $t[0]);

$data[] = $tmp;

endforeach;

echo json\_encode(array("suggestions" => $data));

exit;

}

/\*

Loads the customer edit form

\*/

function view($customer\_id = -1)

{

$data['person\_info'] = $this->Customer->get\_info($customer\_id);

$financial\_infos = "";

if (isset($data['person\_info']->income\_sources))

{

$financial\_infos = json\_decode($data['person\_info']->income\_sources, true);

}

$tmp = array();

if (is\_array($financial\_infos))

{

foreach ($financial\_infos as $financial\_info):

if ($financial\_info !== '=')

{

$tmp[] = explode("=", $financial\_info);

}

endforeach;

}

if (count($tmp) === 0)

{

$tmp[] = array("", "");

}

$attachments = $this->Customer->get\_attachments($customer\_id);

$file = array();

foreach ($attachments as $attachment)

{

$file[] = $this->\_get\_formatted\_file($attachment->attachment\_id, $attachment->filename);

}

$data['attachments'] = $file;

$data['customer\_id'] = $customer\_id;

$data['financial\_infos'] = $tmp;

$this->load->view("customers/form", $data);

}

private function \_get\_formatted\_file($id, $filename)

{

$words = array("doc", "docx", "odt");

$xls = array("xls", "xlsx", "csv");

$tmp = explode(".", $filename);

$ext = $tmp[1];

if (in\_array(strtolower($ext), $words))

{

$tmp['icon'] = "images/word-filetype.jpg";

$tmp['filename'] = $filename;

$tmp['id'] = $id;

}

else if (strtolower($ext) === "pdf")

{

$tmp['icon'] = "images/pdf-filetype.jpg";

$tmp['filename'] = $filename;

$tmp['id'] = $id;

}

else if (in\_array(strtolower($ext), $xls))

{

$tmp['icon'] = "images/xls-filetype.jpg";

$tmp['filename'] = $filename;

$tmp['id'] = $id;

}

else

{

$tmp['icon'] = "images/image-filetype.jpg";

$tmp['filename'] = $filename;

$tmp['id'] = $id;

}

return $tmp;

}

/\*

Inserts/updates a customer

\*/

function save($customer\_id = -1)

{

$person\_data = array(

'first\_name' => $this->input->post('first\_name'),

'last\_name' => $this->input->post('last\_name'),

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

'phone\_number' => $this->input->post('phone\_number'),

'address\_1' => $this->input->post('address\_1'),

'address\_2' => $this->input->post('address\_2'),

'city' => $this->input->post('city'),

'state' => $this->input->post('state'),

'zip' => $this->input->post('zip'),

'country' => $this->input->post('country'),

'comments' => $this->input->post('comments')

);

$customer\_data = array(

'account\_number' => $this->input->post('account\_number') == '' ? null : $this->input->post('account\_number'),

'taxable' => $this->input->post('taxable') == '' ? 0 : 1,

);

if (is\_array($this->input->post("sources")))

{

$income\_sources = array();

$i = 0;

foreach ($this->input->post("sources") as $sources)

{

$tmp = $this->input->post("values");

$income\_sources[] = $sources . "=" . $tmp[$i];

$i++;

}

}

$financial\_data = array(

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

"income\_sources" => json\_encode($income\_sources)

);

if ($this->Customer->save($person\_data, $customer\_data, $customer\_id, $financial\_data))

{

//New customer

if ($customer\_id == -1)

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('customers\_successful\_adding') . ' ' .

$person\_data['first\_name'] . ' ' . $person\_data['last\_name'], 'person\_id' => $customer\_data['person\_id']));

}

else //previous customer

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('customers\_successful\_updating') . ' ' .

$person\_data['first\_name'] . ' ' . $person\_data['last\_name'], 'person\_id' => $customer\_id));

}

}

else//failure

{

echo json\_encode(array('success' => false, 'message' => $this->lang->line('customers\_error\_adding\_updating') . ' ' .

$person\_data['first\_name'] . ' ' . $person\_data['last\_name'], 'person\_id' => -1));

}

}

/\*

This deletes customers from the customers table

\*/

function delete()

{

$customers\_to\_delete = $this->input->post('ids');

if ($this->Customer->delete\_list($customers\_to\_delete))

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('customers\_successful\_deleted') . ' ' .

count($customers\_to\_delete) . ' ' . $this->lang->line('customers\_one\_or\_multiple')));

}

else

{

echo json\_encode(array('success' => false, 'message' => $this->lang->line('customers\_cannot\_be\_deleted')));

}

}

function excel()

{

$data = file\_get\_contents("import\_customers.csv");

$name = 'import\_customers.csv';

force\_download($name, $data);

}

function excel\_import()

{

$this->load->view("customers/excel\_import", null);

}

function do\_excel\_import()

{

$msg = 'do\_excel\_import';

$failCodes = array();

if ($\_FILES['file\_path']['error'] != UPLOAD\_ERR\_OK)

{

$msg = $this->lang->line('items\_excel\_import\_failed');

echo json\_encode(array('success' => false, 'message' => $msg));

return;

}

else

{

if (($handle = fopen($\_FILES['file\_path']['tmp\_name'], "r")) !== FALSE)

{

//Skip first row

fgetcsv($handle);

$i = 1;

while (($data = fgetcsv($handle)) !== FALSE)

{

$person\_data = array(

'first\_name' => $data[0],

'last\_name' => $data[1],

'email' => $data[2],

'phone\_number' => $data[3],

'address\_1' => $data[4],

'address\_2' => $data[5],

'city' => $data[6],

'state' => $data[7],

'zip' => $data[8],

'country' => $data[9],

'comments' => $data[10]

);

$customer\_data = array(

'account\_number' => $data[11] == '' ? null : $data[11],

'taxable' => $data[12] == '' ? 0 : 1,

);

if (!$this->Customer->save($person\_data, $customer\_data))

{

$failCodes[] = $i;

}

$i++;

}

}

else

{

echo json\_encode(array('success' => false, 'message' => 'Your upload file has no data or not in supported format.'));

return;

}

}

$success = true;

if (count($failCodes) > 1)

{

$msg = "Most customers imported. But some were not, here is list of their CODE (" . count($failCodes) . "): " . implode(", ", $failCodes);

$success = false;

}

else

{

$msg = "Import Customers successful";

}

echo json\_encode(array('success' => $success, 'message' => $msg));

}

/\*

get the width for the add/edit form

\*/

function get\_form\_width()

{

return 350;

}

function data()

{

$sel\_user = $this->input->get("employee\_id");

$index = isset($\_GET['order'][0]['column']) ? $\_GET['order'][0]['column'] : 1;

$dir = isset($\_GET['order'][0]['dir']) ? $\_GET['order'][0]['dir'] : "asc";

$order = array("index" => $index, "direction" => $dir);

$length = isset($\_GET['length'])?$\_GET['length']:50;

$start = isset($\_GET['start'])?$\_GET['start']:0;

$key = isset($\_GET['search']['value'])?$\_GET['search']['value']:"";

$people = $this->Customer->get\_all($length, $start, $key, $order, $sel\_user);

$format\_result = array();

foreach ($people->result() as $person)

{

$format\_result[] = array(

"<input type='checkbox' name='chk[]' id='person\_$person->person\_id' value='" . $person->person\_id . "'/>",

$person->person\_id,

$person->last\_name,

$person->first\_name,

$person->email,

$person->phone\_number,

anchor('customers/view/' . $person->person\_id, $this->lang->line('common\_view'), array('class' => 'btn btn-success', "title" => "Update Customer"))

);

}

$data = array(

"recordsTotal" => $this->Customer->count\_all(),

"recordsFiltered" => $this->Customer->count\_all(),

"data" => $format\_result

);

echo json\_encode($data);

exit;

}

function upload\_attachment()

{

$directory = FCPATH . 'uploads/customer-' . $\_REQUEST["customer\_id"] . "/";

$this->load->library('uploader');

$data = $this->uploader->upload($directory);

$this->Customer->save\_attachments($data['params']['customer\_id'], $data);

$file = $this->\_get\_formatted\_file($data['attachment\_id'], $data['filename']);

$file['customer\_id'] = $data['params']['customer\_id'];

echo json\_encode($file);

exit;

}

function customer\_search()

{

$suggestions = $this->Customer->get\_customer\_search\_suggestions($this->input->get('query'), 30);

$data = $tmp = array();

foreach ($suggestions as $suggestion):

$t = explode("|", $suggestion);

$tmp = array("value" => $t[1], "data" => $t[0], "email" => $t[2]);

$data[] = $tmp;

endforeach;

echo json\_encode(array("suggestions" => $data));

exit;

}

}

?>

**-Model:**

<?php

class Customer extends Person {

/\*

Determines if a given person\_id is a customer

\*/

function exists($person\_id, $email = false)

{

$this->db->from('customers');

$this->db->join('people', 'people.person\_id = customers.person\_id');

$this->db->where('customers.person\_id', $person\_id);

if ($email)

{

$this->db->or\_where('people.email', $email);

}

$query = $this->db->get();

return ($query->num\_rows() > 0);

}

/\*

Returns all the customers

\*/

function get\_all($limit = 10000, $offset = 0, $search = "", $order = array(), $sel\_user = false)

{

$sorter = array("", "customers.person\_id", "last\_name", "first\_name", "email", "phone\_number");

$this->db->from('customers');

$this->db->join('people', 'customers.person\_id=people.person\_id');

$this->db->where('deleted', 0);

$employee\_id = $this->session->userdata('person\_id');

$employee\_id = ($sel\_user) ? $sel\_user : $employee\_id;

if ($employee\_id !== "1")

{

$this->db->where("added\_by", $employee\_id);

}

if ($search !== "")

{

$this->db->where('first\_name LIKE ', '%' . $search . '%');

$this->db->or\_where('last\_name LIKE', '%' . $search . '%');

$this->db->or\_where('email LIKE', '%' . $search . '%');

$this->db->or\_where('phone\_number LIKE', '%' . $search . '%');

}

if (count($order) > 0 && $order['index'] < count($sorter))

{

$this->db->order\_by($sorter[$order['index']], $order['direction']);

}

else

{

$this->db->order\_by("last\_name", "asc");

}

$this->db->limit($limit);

$this->db->offset($offset);

return $this->db->get();

}

function count\_all()

{

$this->db->from('customers');

$this->db->where('deleted', 0);

return $this->db->count\_all\_results();

}

/\*

Gets information about a particular customer

\*/

function get\_info($customer\_id)

{

$this->db->from('customers');

$this->db->select('customers.\*');

$this->db->select('people.\*');

$this->db->select('financial\_status.financial\_status\_id');

$this->db->select('financial\_status.income\_sources');

$this->db->join('people', 'people.person\_id = customers.person\_id');

$this->db->join('financial\_status', 'financial\_status.person\_id = people.person\_id', 'LEFT');

$this->db->where('customers.person\_id', $customer\_id);

$query = $this->db->get();

if ($query->num\_rows() == 1)

{

return $query->row();

}

else

{

//Get empty base parent object, as $customer\_id is NOT an customer

$person\_obj = parent::get\_info(-1);

//Get all the fields from customer table

$fields = $this->db->list\_fields('customers');

//append those fields to base parent object, we we have a complete empty object

foreach ($fields as $field)

{

$person\_obj->$field = '';

}

return $person\_obj;

}

}

/\*

Gets information about multiple customers

\*/

function get\_multiple\_info($customer\_ids)

{

$this->db->from('customers');

$this->db->join('people', 'people.person\_id = customers.person\_id');

$this->db->where\_in('customers.person\_id', $customer\_ids);

$this->db->order\_by("last\_name", "asc");

return $this->db->get();

}

function get\_attachments($customer\_id)

{

$this->db->from('attachments');

$this->db->where('customer\_id', $customer\_id);

$query = $this->db->get();

return $query->result();

}

/\*

Inserts or updates a customer

\*/

function save(&$person\_data, &$customer\_data, $customer\_id = false, &$financial\_data = array())

{

$success = false;

//Run these queries as a transaction, we want to make sure we do all or nothing

$this->db->trans\_start();

if (parent::save($person\_data, $customer\_id))

{

$employee\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

if (!$customer\_id or ! $this->exists($customer\_id))

{

$financial\_data['person\_id'] = $person\_data['person\_id'];

$this->db->insert('financial\_status', $financial\_data);

$customer\_data['person\_id'] = $person\_data['person\_id'];

$customer\_data['added\_by'] = $employee\_id;

$success = $this->db->insert('customers', $customer\_data);

$this->move\_attachments($customer\_data);

}

else

{

//$this->db->where('person\_id', $customer\_id);

if ($financial\_data['financial\_status\_id'] > 0)

{

$this->db->where('financial\_status\_id', $financial\_data['financial\_status\_id']);

$this->db->update('financial\_status', $financial\_data);

}

else

{

$financial\_data['person\_id'] = $customer\_id;

$this->db->insert('financial\_status', $financial\_data);

}

$customer\_data['added\_by'] = $employee\_id;

$this->db->where('person\_id', $customer\_id);

$success = $this->db->update('customers', $customer\_data);

}

}

$this->db->trans\_complete();

return $success;

}

function move\_attachments($customer\_data)

{

$linker = $this->session->userdata('linker');

$this->db->from('attachments');

$this->db->where('session\_id', $linker);

$query = $this->db->get();

$this->db->where('session\_id', $linker);

$this->db->update('attachments', array("customer\_id" => $customer\_data['person\_id']));

$attachments = $query->result();

foreach ($attachments as $attachment)

{

$tmp\_dir = FCPATH . "uploads/customer-/";

$user\_dir = FCPATH . "uploads/customer-" . $customer\_data['person\_id'] . "/";

if (!file\_exists($user\_dir))

{

// temporary set to full access

@mkdir($user\_dir);

}

$target\_dist = $user\_dir . $attachment->filename;

if (file\_exists($tmp\_dir . $attachment->filename))

{

copy($tmp\_dir . $attachment->filename, $target\_dist);

unlink($tmp\_dir . $attachment->filename);

}

}

}

/\*

Deletes one customer

\*/

function delete($customer\_id)

{

$this->db->where('person\_id', $customer\_id);

return $this->db->update('customers', array('deleted' => 1));

}

/\*

Deletes a list of customers

\*/

function delete\_list($customer\_ids)

{

$this->db->where\_in('person\_id', $customer\_ids);

return $this->db->update('customers', array('deleted' => 1));

}

/\*

Get search suggestions to find customers

\*/

function get\_search\_suggestions($search, $limit = 25)

{

$suggestions = array();

$this->db->from('customers');

$this->db->join('people', 'customers.person\_id=people.person\_id');

$this->db->where("(first\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

last\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

CONCAT(`first\_name`,' ',`last\_name`) LIKE '%" . $this->db->escape\_like\_str($search) . "%') and deleted=0");

$this->db->order\_by("last\_name", "asc");

$by\_name = $this->db->get();

foreach ($by\_name->result() as $row)

{

$suggestions[] = $row->first\_name . ' ' . $row->last\_name;

}

$this->db->from('customers');

$this->db->join('people', 'customers.person\_id=people.person\_id');

$this->db->where('deleted', 0);

$this->db->like("email", $search);

$this->db->order\_by("email", "asc");

$by\_email = $this->db->get();

foreach ($by\_email->result() as $row)

{

$suggestions[] = $row->email;

}

$this->db->from('customers');

$this->db->join('people', 'customers.person\_id=people.person\_id');

$this->db->where('deleted', 0);

$this->db->like("phone\_number", $search);

$this->db->order\_by("phone\_number", "asc");

$by\_phone = $this->db->get();

foreach ($by\_phone->result() as $row)

{

$suggestions[] = $row->phone\_number;

}

$this->db->from('customers');

$this->db->join('people', 'customers.person\_id=people.person\_id');

$this->db->where('deleted', 0);

$this->db->like("account\_number", $search);

$this->db->order\_by("account\_number", "asc");

$by\_account\_number = $this->db->get();

foreach ($by\_account\_number->result() as $row)

{

$suggestions[] = $row->account\_number;

}

//only return $limit suggestions

if (count($suggestions > $limit))

{

$suggestions = array\_slice($suggestions, 0, $limit);

}

return $suggestions;

}

/\*

Get search suggestions to find customers

\*/

function get\_customer\_search\_suggestions($search, $limit = 25)

{

$suggestions = array();

$user\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

$add\_where = '';

if ($user\_id > 1)

{

$add\_where = " and added\_by = " . $user\_id . " ";

}

$this->db->from('customers');

$this->db->join('people', 'customers.person\_id=people.person\_id');

$this->db->where("(first\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

last\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

CONCAT(`first\_name`,' ',`last\_name`) LIKE '%" . $this->db->escape\_like\_str($search) . "%') and deleted=0

" . $add\_where . "

");

$this->db->order\_by("last\_name", "asc");

$by\_name = $this->db->get();

foreach ($by\_name->result() as $row)

{

$suggestions[] = $row->person\_id . '|' . $row->first\_name . ' ' . $row->last\_name . '|' . $row->email;

}

$this->db->from('customers');

$this->db->join('people', 'customers.person\_id=people.person\_id');

$this->db->where('deleted', 0);

$this->db->like("account\_number", $search);

$this->db->order\_by("account\_number", "asc");

$by\_account\_number = $this->db->get();

foreach ($by\_account\_number->result() as $row)

{

$suggestions[] = $row->person\_id . '|' . $row->account\_number . '|' . $row->email;

}

//only return $limit suggestions

if (count($suggestions > $limit))

{

$suggestions = array\_slice($suggestions, 0, $limit);

}

return $suggestions;

}

/\*

Preform a search on customers

\*/

function search($search)

{

$this->db->from('customers');

$this->db->join('people', 'customers.person\_id=people.person\_id');

$this->db->where("(first\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

last\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

email LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

phone\_number LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

account\_number LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

CONCAT(`first\_name`,' ',`last\_name`) LIKE '%" . $this->db->escape\_like\_str($search) . "%') and deleted=0");

$this->db->order\_by("last\_name", "asc");

return $this->db->get();

}

function save\_attachments($customer\_id, &$data)

{

if ($customer\_id > 0)

{

if ($this->db->insert('attachments', array("filename" => $data['filename'], "customer\_id" => $customer\_id)))

{

$data['attachment\_id'] = $this->db->insert\_id();

return true;

}

}

$session\_id = $data['params']['linker'];

$this->load->library('session');

$this->session->set\_userdata(array("linker" => $session\_id));

if ($this->db->insert('attachments', array("filename" => $data['filename'], "session\_id" => $session\_id)))

{

$data['attachment\_id'] = $this->db->insert\_id();

return true;

}

}

}

?**>**

**-View:**

<?php $this->load->view("partial/header"); ?>

<?= form\_open('customers/save/' . $person\_info->person\_id, array('id' => 'customer\_form', 'class' => 'form-horizontal')); ?>

<input type="hidden" id="customer\_id" value="<?= $customer\_id ?>" />

<input type="hidden" id="linker" value="<?= random\_string('alnum', 16); ?>" />

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-content">

<h2><?php echo $this->lang->line('common\_list\_of') . ' ' . $this->lang->line('module\_' . $controller\_name); ?></h2>

<ol class="breadcrumb">

<li>

<a href="<?= site\_url(); ?>">Home</a>

</li>

<li>

<a href="<?= site\_url("customers"); ?>"><?= ucwords($this->lang->line('module\_' . $controller\_name)); ?></a>

</li>

<li class="active">

<strong>Add</strong>

</li>

</ol>

</div>

</div>

</div>

</div>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-title">

<h5>

<?php echo $this->lang->line("customers\_basic\_information"); ?>

</h5>

<div class="inqbox-tools">

<a class="collapse-link">

<i class="fa fa-chevron-up"></i>

</a>

<a class="close-link">

<i class="fa fa-times"></i>

</a>

</div>

</div>

<div class="inqbox-content">

<div class="tabs-container">

<ul class="nav nav-tabs tab-border-top-danger">

<li class="active"><a data-toggle="tab" href="#sectionA"><?= $this->lang->line("customers\_personal\_information"); ?></a></li>

<li><a data-toggle="tab" href="#sectionB"><?= $this->lang->line("customers\_financial\_information"); ?></a></li>

<li><a data-toggle="tab" href="#sectionC"><?= $this->lang->line("customers\_attachment"); ?></a></li>

</ul>

<div class="tab-content">

<div id="sectionA" class="tab-pane fade in active">

<div style="text-align: center">

<div id="required\_fields\_message"><?php echo $this->lang->line('common\_fields\_required\_message'); ?></div>

<ul id="error\_message\_box"></ul>

</div>

<?php $this->load->view("people/form\_basic\_info"); ?>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('customers\_account\_number') . ':', 'account\_number'); ?></label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'account\_number',

'id' => 'account\_number',

'value' => $person\_info->account\_number,

'class' => 'form-control'

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('customers\_taxable') . ':', 'taxable'); ?></label>

<div class="col-sm-10">

<?php echo form\_checkbox('taxable', '1', $person\_info->taxable == '' ? TRUE : (boolean) $person\_info->taxable); ?>

</div>

</div>

<div class="hr-line-dashed"></div>

</div>

<div id="sectionB" class="tab-pane fade">

<input type="hidden" name="financial\_status\_id" value="<?= @$person\_info->financial\_status\_id; ?>" />

<table class="table" id="tbl-income-sources">

<thead>

<tr>

<th style="text-align: center; width: 1%">

<input type="checkbox" class="select\_all\_" />

</th>

<th style="text-align: center; width: 80%"><?= $this->lang->line("customers\_occupation"); ?></th>

<th style="text-align: center; width: 20%"><?= $this->lang->line("customers\_monthly\_income"); ?></th>

</tr>

</thead>

<tbody>

<?php foreach ($financial\_infos as $financial\_info): ?>

<tr>

<td>

<input type="checkbox" class="select\_" />

</td>

<td>

<input type="text" class="form-control" name="sources[]" value="<?= $financial\_info[0]; ?>" />

</td>

<td>

<input type="number" class="form-control" step="any" name="values[]" value="<?= $financial\_info[1]; ?>" />

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<button class="btn btn-primary" type="button" id="btn-add-row"><?= $this->lang->line('common\_add\_row'); ?></button>

<button class="btn btn-danger" type="button" id="btn-del-row"><?= $this->lang->line('common\_delete\_row'); ?></button>

</div>

<div id="sectionC" class="tab-pane fade">

<h3><?= $this->lang->line('loans\_attachments') ?></h3>

<div id="required\_fields\_message"><?php echo $this->lang->line('loans\_attachments\_message'); ?></div>

<div>

<ul class="list-inline" id="filelist\_a">

<?php foreach ($attachments as $attachment): ?>

<li>

<a href="uploads/customer-<?= $person\_info->person\_id; ?>/<?= $attachment['filename']; ?>" target="\_blank" title="<?= $attachment['filename']; ?>"><img src="<?= $attachment['icon']; ?>" /></a>

<span class="close remove-file" data-file-id="<?= $attachment['id']; ?>" title="Remove this file"><i class="fa fa-times-circle"></i></span>

</li>

<?php endforeach; ?>

</ul>

</div>

<div id="progress\_a" class="overlay"></div>

<div class="progress progress-task" style="height: 4px; width: 15%; margin-bottom: 2px; display: none">

<div style="width: 0%;" aria-valuemax="100" aria-valuemin="0" aria-valuenow="50" role="progressbar" class="progress-bar progress-bar-info">

</div>

</div>

<div id="container\_a">

<a id="pickfiles\_a" href="javascript:;" class="btn btn-default" data-customer-id="<?= $person\_info->person\_id; ?>"><?= $this->lang->line('common\_browse'); ?></a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="form-group">

<div class="col-sm-4 col-sm-offset-2">

<button type="button" class="btn btn-default" data-dismiss="modal" id="btn-close"><?= $this->lang->line("common\_close"); ?></button>

<?php if ((int) $customer\_id > -1) : ?>

<button type="button" class="btn btn-primary" id="btn-edit"><?= $this->lang->line("common\_edit"); ?></button>

<?php endif; ?>

<?php

$display = '';

if ($customer\_id > -1)

{

$display = 'display: none';

}

echo form\_submit(

array(

'name' => 'submit',

'id' => 'btn-save',

'value' => $this->lang->line('common\_save'),

'class' => 'btn btn-primary',

'style' => $display

)

);

?>

</div>

</div>

</div>

<?= form\_close(); ?>

<?php $this->load->view("partial/footer"); ?>

<script src="<?php echo base\_url(); ?>js/people.js?v=<?= time(); ?>"></script>

<script type="text/javascript">

$(document).ready(function () {

if ($("#customer\_id").val() > -1)

{

$("#customer\_form input, textarea").prop("disabled", true);

$("#btn-edit").click(function () {

$("#btn-save").show();

$(this).hide();

$("#customer\_form input, textarea").prop("disabled", false);

});

}

var settings = {

submitHandler: function (form) {

$("#submit").prop("disabled", true);

$(form).ajaxSubmit({

success: function (response) {

post\_person\_form\_submit(response);

$("#submit").prop("disabled", false);

},

dataType: 'json',

type: 'post'

});

},

rules: {

first\_name: "required",

last\_name: "required",

email: "email"

},

messages: {

first\_name: "<?php echo $this->lang->line('common\_first\_name\_required'); ?>",

last\_name: "<?php echo $this->lang->line('common\_last\_name\_required'); ?>",

email: "<?php echo $this->lang->line('common\_email\_invalid\_format'); ?>"

}

};

$('#customer\_form').validate(settings);

function post\_person\_form\_submit(response)

{

if (!response.success)

{

set\_feedback(response.message, 'error\_message', true);

}

else

{

set\_feedback(response.message, 'success\_message', false);

}

$("#customer\_form").attr("action", "<?= site\_url(); ?>customers/save/" + response.person\_id);

}

});

</script>

<?php

echo form\_open\_multipart('customers/do\_excel\_import/',array('id'=>'item\_form'));

?>

<div id="required\_fields\_message">Import customers from Excel sheet</div>

<ul id="error\_message\_box"></ul>

<b><a href="<?php echo site\_url('customers/excel'); ?>">Download Import Excel Template (CSV)</a></b>

<fieldset id="item\_basic\_info">

<legend>Import</legend>

<div class="field\_row clearfix">

<?php echo form\_label('File path:', 'name',array('class'=>'wide')); ?>

<div class='form\_field'>

<?php echo form\_upload(array(

'name'=>'file\_path',

'id'=>'file\_path',

'value'=>'')

);?>

</div>

</div>

<?php

echo form\_submit(array(

'name'=>'submitf',

'id'=>'submitf',

'value'=>$this->lang->line('common\_submit'),

'class'=>'submit\_button float\_right')

);

?>

</fieldset>

<?php

echo form\_close();

?>

<script type='text/javascript'>

//validation and submit handling

$(document).ready(function()

{

$('#item\_form').validate({

submitHandler:function(form)

{

$(form).ajaxSubmit({

success:function(response)

{

tb\_remove();

post\_person\_form\_submit(response);

},

dataType:'json'

});

},

errorLabelContainer: "#error\_message\_box",

wrapper: "li",

rules:

{

file\_path:"required"

},

messages:

{

file\_path:"Full path to excel file required"

}

});

});

</script>

**3.Roles**

**-Controller:**

<?php

require\_once ("secure\_area.php");

require\_once ("interfaces/idata\_controller.php");

class Roles extends Secure\_area implements iData\_controller {

private $\_model;

private $\_model\_name;

function \_\_construct()

{

parent::\_\_construct(strtolower(get\_class()));

$model\_name = ucfirst(str\_replace("s", "", get\_class()));

$this->load->model($model\_name);

$this->\_model = $this->$model\_name;

$this->\_model\_name = strtolower($model\_name);

}

function index()

{

$data['controller\_name'] = strtolower(get\_class());

$res = $this->Employee->getLowerLevels();

$data['staffs'] = $res;

// just make sure that controller name is equivalent to the table name in database

$data['fields'] = $this->\_model->get\_fields($data['controller\_name']);

$this->load->view($data['controller\_name'] . '/manage', $data);

}

function view($id = -1)

{

$controller\_name = strtolower(get\_class());

$data['info'] = $this->\_model->get\_info($id);

$data['low\_levels'] = json\_decode($data['info']->low\_level, TRUE);

$data['low\_levels'] = is\_array($data['low\_levels']) ? $data['low\_levels'] : array();

$data['module\_ids'] = json\_decode($data["info"]->rights, TRUE);

$data['module\_ids'] = is\_array($data['module\_ids']) ? $data['module\_ids'] : array();

$data['id'] = $id;

$data['roles'] = $this->\_model->get\_all\_roles();

$data['all\_modules'] = $this->Module->get\_all\_modules();

$data['all\_subpermissions'] = $this->Module->get\_all\_subpermissions();

$data['controller\_name'] = $controller\_name;

$this->load->view($controller\_name . "/form", $data);

}

function save($id = -1)

{

$data["role\_id"] = $id;

$data["name"] = $this->input->post("role\_name");

$data["low\_level"] = json\_encode($this->input->post("low\_level"));

$data["rights"] = json\_encode($this->input->post("rights"));

if ($this->\_model->save($data, $id))

{

//New

if ($id == -1)

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line(strtolower(get\_class()) . '\_successful\_adding') . ' ' .

$data[$this->\_model\_name . '\_id'], $this->\_model\_name . '\_id' => $data[$this->\_model\_name . '\_id']));

$id = $data[$this->\_model\_name . '\_id'];

}

else //previous item

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line(strtolower(get\_class()) . '\_successful\_updating') . ' ' .

$data[$this->\_model\_name . '\_id'], $this->\_model\_name . '\_id' => $id));

}

}

else//failure

{

echo json\_encode(array('success' => false, 'message' => $this->lang->line(strtolower(get\_class()) . '\_error\_adding\_updating') . ' ' .

$data[$this->\_model\_name . '\_id'], $this->\_model\_name . '\_id' => -1));

}

}

function delete()

{

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

$controller\_name = strtolower(get\_class());

if ($this->\_model->delete\_list($ids))

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line($controller\_name . '\_successful\_deleted') . ' ' .

count($ids) . ' ' . $this->lang->line($controller\_name . '\_one\_or\_multiple')));

}

else

{

echo json\_encode(array('success' => false, 'message' => $this->lang->line('loan\_type\_cannot\_be\_deleted')));

}

}

function data()

{

$index = isset($\_GET['order'][0]['column']) ? $\_GET['order'][0]['column'] : 1;

$dir = isset($\_GET['order'][0]['dir']) ? $\_GET['order'][0]['dir'] : "asc";

$order = array("index" => $index, "direction" => $dir);

$length = isset($\_GET['length']) ? $\_GET['length'] : 50;

$start = isset($\_GET['start']) ? $\_GET['start'] : 0;

$key = isset($\_GET['search']['value']) ? $\_GET['search']['value'] : "";

$controller\_name = strtolower(get\_class());

$$controller\_name = $this->\_model->get\_all($length, $start, $key, $order);

$id = strtolower(str\_replace("s", "", $controller\_name) . "\_id");

$format\_result = array();

$fields = $this->\_model->get\_fields(ucfirst($controller\_name));

foreach ($$controller\_name->result() as $data)

{

$tmp = array();

$tmp[] = "<input type='checkbox' name='chk[]' class='select\_' id='" . $controller\_name . "\_" . $data->$id . "' value='" . $data->$id . "'/>";

foreach ($fields as $field)

{

if ($field === 'low\_level')

{

if ($data->$field !== "false")

{

$tmp[] = $this->\_model->getStaffs(json\_decode($data->$field, TRUE));

}

else

{

$tmp[] = "";

}

}

else if ($field === 'rights')

{

if ($data->$field !== "false")

{

$tmp[] = implode(", ",json\_decode($data->$field, TRUE));

}

else

{

$tmp[] = "";

}

}

else if($field === 'added\_by')

{

$emp = $this->Employee->get\_info($data->$field);

$tmp[] = (isset($emp->first\_name) ? ucwords($emp->first\_name." ".$emp->last\_name) : "");

}

else

{

$tmp[] = $data->$field;

}

}

$action = array();

$action[] = anchor($controller\_name . '/view/' . $data->$id, $this->lang->line('common\_edit'), array('class' => 'btn btn-success', "title" => $this->lang->line($controller\_name . '\_update')));

$tmp[] = implode("&nbsp", $action);

$format\_result[] = $tmp;

}

$data = array(

"recordsTotal" => $this->\_model->count\_all(),

"recordsFiltered" => $this->\_model->count\_all(),

"data" => $format\_result

);

echo json\_encode($data);

exit;

}

public function get\_row()

{

}

public function search()

{

}

public function suggest()

{

}

public function get\_form\_width()

{

}

public function upload()

{

$directory = FCPATH . "modules/";

$this->load->library('uploader');

$data = $this->uploader->upload($directory);

$module\_id = $this->\_model->add\_plugin($data);

// decompress the archive file

$decompress = $this->\_decompress($data['filename']);

echo json\_encode(array("filename" => $data['filename'], "module\_id" => $module\_id, "decompress" => $decompress));

exit;

}

private function \_decompress($file)

{

// get the absolute path to $file

$path = FCPATH . "modules/";

$file = FCPATH . "modules/" . $file;

$zip = new ZipArchive;

$res = $zip->open($file);

if ($res === TRUE)

{

// extract it to the path we determined above

$zip->extractTo($path);

$zip->close();

return true;

}

return false;

}

public function activate($id)

{

$data = $this->\_model->get\_plugin($id);

$files = json\_decode($data->module\_files, true);

$plugin\_name = $data->module\_name;

$controllers = $files['controllers'];

$this->\_move\_files($controllers, 'controllers', $plugin\_name);

$models = $files['models'];

$this->\_move\_files($models, 'models', $plugin\_name);

$views = $files['views'];

$this->\_move\_files($views, 'views', $plugin\_name);

$language = $files['language'];

$this->\_move\_files($language, 'language', $plugin\_name);

$js = $files['js'];

$this->\_move\_files($js, 'js', $plugin\_name);

// finally, let's save to database;

$this->\_model->register\_plugin($plugin\_name);

// update status as active

$this->\_model->update\_status("active", $id);

//var\_dump($data);

}

private function \_move\_files($arr, $type, $plugin\_name)

{

foreach ($arr as $path)

{

if (strpos($path, '\*') !== false)

{

$src = 'modules/' . $plugin\_name;

$dest = 'application';

$files = glob(FCPATH . "modules/" . $plugin\_name . str\_replace("\*", "", $path) . "\*.\*");

foreach ($files as $file)

{

$file\_to\_go = str\_replace($src, $dest, $file);

$dir = str\_replace($src, $dest, dirname($file));

if (!file\_exists($dir))

{

@mkdir($dir);

}

copy($file, $file\_to\_go);

}

}

}

}

}

?>

**-Model:**

<?php

class Role extends CI\_Model {

/\*

Determines if a given id exist

\*/

function exists($id)

{

$this->db->from(strtolower(get\_class()) . 's');

$this->db->where(strtolower(get\_class()) . '\_id', $id);

$query = $this->db->get();

return ($query->num\_rows() == 1);

}

function get\_fields($controller\_name)

{

return $this->db->list\_fields(strtolower($controller\_name));

}

function get\_all($limit = 10000, $offset = 0, $search = "", $order = array())

{

$user\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

$sorter = $this->get\_fields(strtolower(get\_class()) . 's');

$this->db->from(strtolower(get\_class()) . 's');

if ($search !== "")

{

// customization needed

$this->db->where($sorter[1] . ' LIKE ', '%' . $search . '%');

}

if (isset($\_GET['employee\_id']) && $\_GET['employee\_id'] > 0)

{

$user\_id = $\_GET['employee\_id'];

}

if ($user\_id !== '1')

{

$this->db->where('added\_by', $user\_id);

}

if (count($order) > 0 && $order['index'] < count($sorter))

{

$this->db->order\_by($sorter[$order['index']], $order['direction']);

}

else

{

// customization needed

$this->db->order\_by(strtolower(get\_class()) . "s\_id", "asc");

}

$this->db->limit($limit);

$this->db->offset($offset);

return $this->db->get();

}

function get\_all\_roles($low\_levels = false)

{

$user\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

if ($low\_levels)

{

$this->db->where\_in("role\_id", $low\_levels);

}

if ($user\_id !== '1')

{

$this->db->where('added\_by', $user\_id);

}

$query = $this->db->get("roles");

return $query->result();

}

function count\_all()

{

$this->db->from(strtolower(get\_class()) . 's');

return $this->db->count\_all\_results();

}

function get\_multiple($ids = -1)

{

$this->db->from(strtolower(get\_class()) . 's');

if ($ids > -1)

{

$this->db->where\_in(strtolower(get\_class()) . '\_id', $ids);

}

return $this->db->get()->result();

}

/\*

Gets information about a particular item kit

\*/

function get\_info($id)

{

$this->db->from(strtolower(get\_class()) . 's');

$this->db->where(strtolower(get\_class()) . '\_id', $id);

$query = $this->db->get();

if ($query->num\_rows() == 1)

{

return $query->row();

}

else

{

//Get empty base parent object, as $item\_kit\_id is NOT an item kit

$item\_obj = new stdClass();

//Get all the fields from items table

$fields = $this->db->list\_fields(strtolower(get\_class()) . 's');

foreach ($fields as $field)

{

$item\_obj->$field = '';

}

return $item\_obj;

}

}

/\*

Inserts or updates an item kit

\*/

function save(&$data, $id = false)

{

$user\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

$data['added\_by'] = $user\_id;

if (!$id or ! $this->exists($id))

{

unset($data["role\_id"]);

if ($this->db->insert(strtolower(get\_class()) . 's', $data))

{

$data[strtolower(get\_class()) . '\_id'] = $this->db->insert\_id();

return true;

}

return false;

}

$this->db->where(strtolower(get\_class()) . '\_id', $id);

return $this->db->update(strtolower(get\_class()) . 's', $data);

}

/\*

Deletes one item

\*/

function delete($id)

{

// though customization if you wish to just have a soft delete

return $this->db->delete(strtolower(get\_class()) . 's', array(strtolower(get\_class()) . '\_id' => $id));

}

/\*

Deletes a list of item kits

\*/

function delete\_list($ids)

{

// though customization if you wish to just have a soft delete

$this->db->where\_in(strtolower(get\_class()) . '\_id', $ids);

return $this->db->delete(strtolower(get\_class()) . 's');

}

/\*

\* Function to remove all files related to this plugin

\*/

function remove\_files($ids)

{

}

function add\_plugin()

{

// parse or decompress the zip file

}

function validate\_plugin($plugin)

{

$plugin\_name = $plugin['plugin']['name'];

$plugin\_desc = isset($plugin['plugin']['description']) ? $plugin['plugin']['description'] : "No description!";

if (!$this->\_exists($plugin\_name))

{

$data['module\_name'] = $plugin\_name;

$data['module\_desc'] = $plugin\_desc;

$data['module\_files'] = json\_encode($plugin['plugin']['files']);

$this->db->insert(strtolower(get\_class()) . 's', $data);

}

}

private function \_exists($name)

{

$this->db->from(strtolower(get\_class()) . 's');

$this->db->where('module\_name', $name);

$query = $this->db->get();

return ($query->num\_rows() == 1);

}

public function get\_plugin($id)

{

$this->db->where("plugin\_id", $id);

$this->db->from(strtolower(get\_class()) . "s");

$query = $this->db->get();

return $query->row();

}

public function register\_plugin($plugin\_name)

{

if (!$this->\_module\_exists($plugin\_name))

{

$data = array();

$data['module\_id'] = $plugin\_name;

$data['name\_lang\_key'] = "module\_" . $plugin\_name;

$data['desc\_lang\_key'] = "module\_" . $plugin\_name . "\_desc";

$data['icons'] = '<i class="fa fa-smile-o" style="font-size: 50px; color:#FF5400"></i>';

$data['is\_active'] = 1;

$this->db->insert('modules', $data);

$data = array();

$data['permission\_id'] = $plugin\_name;

$data['module\_id'] = $plugin\_name;

$this->db->insert('permissions', $data);

$data = array();

$data['permission\_id'] = $plugin\_name;

$data['person\_id'] = 1;

$this->db->insert('grants', $data);

// execute the file table

$files = glob(FCPATH . "modules/" . $plugin\_name . "/sql/\*.\*");

foreach ($files as $file)

{

$this->db->query(file\_get\_contents($file));

}

}

}

private function \_module\_exists($plugin\_name)

{

$this->db->from("modules");

$this->db->where("module\_id", $plugin\_name);

$query = $this->db->get();

return ($query->num\_rows() == 1);

}

public function update\_status($status, $id)

{

$this->db->where("plugin\_id", $id);

$this->db->update(strtolower(get\_class()).'s', array("status\_flag" => $status));

}

public function getStaffs($ids)

{

$this->db->where\_in("role\_id", $ids);

$results = $this->db->get("roles")->result();

$tmp = array();

foreach($results as $result):

$tmp[] = $result->name;

endforeach;

return implode(", ", $tmp);

}

}

?>

**-View:**

<?php $this->load->view("partial/header"); ?>

<?php echo form\_open($controller\_name . '/save/' . $id, array('id' => $controller\_name . '\_form', 'class' => 'form-horizontal')); ?>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-content">

<h2><?php echo $this->lang->line('common\_list\_of') . ' ' . $this->lang->line('module\_' . $controller\_name); ?></h2>

<ol class="breadcrumb">

<li>

<a href="<?= site\_url(); ?>">Home</a>

</li>

<li>

<a href="<?= site\_url("roles"); ?>"><?= ucwords($this->lang->line('module\_' . $controller\_name)); ?></a>

</li>

<li class="active">

<strong>Add</strong>

</li>

</ol>

</div>

</div>

</div>

</div>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-title">

<h5>

<?php echo $this->lang->line("info"); ?>

</h5>

<div class="inqbox-tools">

<a class="collapse-link">

<i class="fa fa-chevron-up"></i>

</a>

<a class="close-link">

<i class="fa fa-times"></i>

</a>

</div>

</div>

<div class="inqbox-content">

<div style="text-align: center">

<div id="required\_fields\_message"><?php echo $this->lang->line('common\_fields\_required\_message'); ?></div>

<ul id="error\_message\_box"></ul>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label('Role Name' . ':', 'role\_name', array('class' => 'required')); ?>

</label>

<div class="col-sm-10">

<input type="text" name="role\_name" value="<?= $info->name; ?>" id="role\_name" class="form-control">

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label('Lower level' . ':', 'low\_level'); ?>

</label>

<div class="col-sm-10">

<ul>

<?php foreach ($roles as $role): ?>

<li style="list-style: none;">

<label style="width: 100%; line-height: 1">

<input type="checkbox" name="low\_level[]" value="<?= $role->role\_id; ?>" <?= in\_array($role->role\_id, $low\_levels) ? "checked='checked'" : ""; ?> />

<span><?= $role->name; ?></span>

</label>

</li>

<?php endforeach; ?>

</ul>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

Permissions

</label>

<div class="col-sm-10">

<fieldset id="access\_rights">

<div style="color:red"><?php echo $this->lang->line("employees\_permission\_desc"); ?></div>

<ul id="permission\_list">

<?php foreach ($all\_modules->result() as $module) : ?>

<li style="list-style: none;">

<label style="width: 100%">

<input type="checkbox" name="rights[]" value="<?= $module->module\_id; ?>" <?= in\_array($module->module\_id, $module\_ids) ? "checked='checked'" : ""; ?>>

<span class="medium"><?php echo $this->lang->line('module\_' . $module->module\_id); ?>:</span>

<span class="small"><?php echo $this->lang->line('module\_' . $module->module\_id . '\_desc'); ?></span>

</label>

</li>

<?php endforeach; ?>

</ul>

</fieldset>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="form-group">

<div class="col-sm-4 col-sm-offset-2">

<button type="button" class="btn btn-default" data-dismiss="modal" id="btn-close"><?= $this->lang->line("common\_close"); ?></button>

<?php

echo form\_submit(

array(

'name' => 'submit',

'id' => 'submit',

'value' => $this->lang->line('common\_submit'),

'class' => 'btn btn-primary'

)

);

?>

</div>

</div>

</div>

<?php

echo form\_close();

?>

<?php $this->load->view("partial/footer"); ?>

<script type='text/javascript'>

//validation and submit handling

$(document).ready(function ()

{

$("#div-form").height($(window).height() - 250);

$('#<?= $controller\_name; ?>\_form').validate({

submitHandler: function (form)

{

$(form).ajaxSubmit({

success: function (response)

{

if (!response.success)

{

set\_feedback(response.message, 'error\_message', true);

} else

{

set\_feedback(response.message, 'success\_message', false);

}

$("#roles\_form").attr("action", "<?= site\_url(); ?>loan\_types/save/" + response.role\_id);

},

dataType: 'json'

});

},

errorLabelContainer: "#error\_message\_box",

wrapper: "li",

rules:

{

name: "required"

},

messages:

{

name: "<?php echo $this->lang->line($controller\_name . '\_name\_required'); ?>",

}

});

});

</script>

<?php $this->load->view("partial/header"); ?>

<style>

table#datatable td:nth-child(6) {

width: 15%;

text-align: center;

white-space: nowrap;

}

table#datatable td:nth-child(7) {

width: 5%;

white-space: nowrap;

}

</style>

<div class="hide-staff" style="display: none;">

<select class="form-control input-sm hidden-xs" id="sel-staff">

<option value="0">Select staff</option>

<?php foreach ($staffs as $staff): ?>

<option value="<?= $staff->person\_id; ?>" <?= ((isset($\_GET['employee\_id'])) && $\_GET['employee\_id'] === $staff->person\_id) ? 'selected="selected"' : ""; ?>><?= $staff->first\_name . " " . $staff->last\_name; ?></option>

<?php endforeach; ?>

</select>

</div>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-content">

<h2><?php echo $this->lang->line('common\_list\_of') . ' ' . $this->lang->line('module\_' . $controller\_name); ?></h2>

<ol class="breadcrumb">

<li>

<a href="<?= site\_url(); ?>">Home</a>

</li>

<li>

<a>Roles</a>

</li>

<li class="active">

<strong>List</strong>

</li>

</ol>

</div>

</div>

</div>

</div>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-title border-top-danger">

<h5>

<?= $this->lang->line('module\_' . $controller\_name); ?>

</h5>

<div class="inqbox-tools">

<?= anchor("$controller\_name/delete", $this->lang->line("common\_delete"), array('id' => 'delete', 'class' => 'btn btn-primary btn-xs', 'style' => 'color:white')); ?>

<?= anchor("$controller\_name/view/-1", "<div class='btn btn-primary btn-xs' style='float: left; margin-right:10px;'><span>" . $this->lang->line($controller\_name . '\_new') . "</span></div>"); ?>

</div>

</div>

<div class="inqbox-content table-responsive">

<table id="datatable" class="table table-hover table-bordered" cellspacing="0" width="100%">

<thead>

<tr>

<th style="text-align: center; width: 1%"><input type="checkbox" class="select\_all\_" /></th>

<?php foreach ($fields as $field): ?>

<th style="text-align: center"><?= $this->lang->line($field) ?></th>

<?php endforeach; ?>

<th style="text-align: center; width: 1%"><?= $this->lang->line("common\_action"); ?></th>

</tr>

</thead>

</table>

</div>

</div>

</div>

</div>

<div id="feedback\_bar"></div>

<?php $this->load->view("partial/footer"); ?>

<script type="text/javascript">

$(document).ready(function ()

{

$('body').on('hidden.bs.modal', '.modal', function () {

$(this).removeData('bs.modal');

});

$("#datatable").dataTable({

"aoColumnDefs": [

{'bSortable': false, 'aTargets': [0, 3]}

],

"language": {

"url": "<?php echo base\_url($this->config->item('language') . ".json"); ?>"

},

"processing": true,

"serverSide": true,

"aLengthMenu": [[50, 100, 200, 100000], [50, 100, 200, "<?= $this->lang->line("common\_all"); ?>"]],

"iDisplayLength": 50,

"order": [1, "desc"],

"ajax": {

"url": "<?php echo site\_url("$controller\_name/data") ?>",

data: {employee\_id: '<?= isset($\_GET['employee\_id']) ? $\_GET['employee\_id'] : false; ?>'},

type: 'get'

},

"initComplete": function (settings, json) {

$("#datatable\_filter").find("input[type='search']").attr("placeholder", "<?= $this->lang->line("common\_search") ?>");

var el = $(".dataTables\_filter").find('label');

el.append("&nbsp;");

el.append($(".hide-staff").html());

}

});

$(document).on("change", "#sel-staff", function(){

location.href = "<?=site\_url($this->uri->segment(1))?>?employee\_id=" + $(this).val();

});

enable\_delete('<?php echo $this->lang->line($controller\_name . "\_confirm\_delete") ?>', '<?php echo $this->lang->line($controller\_name . "\_none\_selected") ?>');

$(".select\_all\_").click(function () {

if ($(this).is(":checked"))

{

$(".select\_").prop("checked", true);

} else

{

$(".select\_").prop("checked", false);

}

});

});

</script>

**4.Employees**

**-Controller**

<?php

require\_once ("person\_controller.php");

class Employees extends Person\_controller {

function \_\_construct()

{

parent::\_\_construct('employees');

$this->load->model("Role");

}

function index()

{

$config['base\_url'] = site\_url('/employees/index');

$config['total\_rows'] = $this->Employee->count\_all();

$config['per\_page'] = '20';

$config['uri\_segment'] = 3;

$this->pagination->initialize($config);

$res = $this->Employee->getLowerLevels();

$data['staffs'] = $res;

$data["has\_lower\_level"] = $this->Employee->hasLowerLevel();

$data['controller\_name'] = strtolower(get\_class());

$data['form\_width'] = $this->get\_form\_width();

$data['manage\_table'] = get\_people\_manage\_table($this->Employee->get\_all($config['per\_page'], $this->uri->segment($config['uri\_segment'])), $this);

$this->load->view('people/employee', $data);

}

/\*

Returns employee table data rows. This will be called with AJAX.

\*/

function search()

{

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

$data\_rows = get\_people\_manage\_table\_data\_rows($this->Employee->search($search), $this);

echo $data\_rows;

}

/\*

Gives search suggestions based on what is being searched for

\*/

function suggest()

{

$suggestions = $this->Employee->get\_search\_suggestions($this->input->post('q'), $this->input->post('limit'));

echo implode("\n", $suggestions);

}

/\*

Loads the employee edit form

\*/

function view($employee\_id = -1)

{

$data['person\_info'] = $this->Employee->get\_info($employee\_id);

$data['all\_modules'] = $this->Module->get\_all\_modules();

$data['all\_subpermissions'] = $this->Module->get\_all\_subpermissions();

$role\_info = $this->Role->get\_info( $this->Employee->get\_logged\_in\_employee\_info()->role\_id );

$low\_levels = $role\_info->low\_level !== "false" ? json\_decode($role\_info->low\_level) : false;

$data['roles'] = $this->Role->get\_all\_roles($low\_levels);

$this->load->view("employees/form", $data);

}

/\*

Inserts/updates an employee

\*/

function save($employee\_id = -1)

{

$person\_data = array(

'first\_name' => $this->input->post('first\_name'),

'last\_name' => $this->input->post('last\_name'),

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

'phone\_number' => $this->input->post('phone\_number'),

'address\_1' => $this->input->post('address\_1'),

'address\_2' => $this->input->post('address\_2'),

'city' => $this->input->post('city'),

'state' => $this->input->post('state'),

'zip' => $this->input->post('zip'),

'country' => $this->input->post('country'),

'comments' => $this->input->post('comments'),

'role\_id' => $this->input->post('role\_id')

);

$grants\_data = json\_decode($this->Role->get\_info($this->input->post("role\_id"))->rights, TRUE );

//Password has been changed OR first time password set

if ($this->input->post('password') != '')

{

$employee\_data = array(

'username' => $this->input->post('username'),

'password' => md5($this->input->post('password'))

);

}

else //Password not changed

{

$employee\_data = array('username' => $this->input->post('username'));

}

if ($this->Employee->save($person\_data, $employee\_data, $grants\_data, $employee\_id))

{

//New employee

if ($employee\_id == -1)

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('employees\_successful\_adding') . ' ' .

$person\_data['first\_name'] . ' ' . $person\_data['last\_name'], 'person\_id' => $employee\_data['person\_id']));

}

else //previous employee

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('employees\_successful\_updating') . ' ' .

$person\_data['first\_name'] . ' ' . $person\_data['last\_name'], 'person\_id' => $employee\_id));

}

}

else//failure

{

echo json\_encode(array('success' => false, 'message' => $this->lang->line('employees\_error\_adding\_updating') . ' ' .

$person\_data['first\_name'] . ' ' . $person\_data['last\_name'], 'person\_id' => -1));

}

}

/\*

This deletes employees from the employees table

\*/

function delete()

{

$employees\_to\_delete = $this->input->post('ids');

if ($this->Employee->delete\_list($employees\_to\_delete))

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('employees\_successful\_deleted') . ' ' .

count($employees\_to\_delete) . ' ' . $this->lang->line('employees\_one\_or\_multiple')));

}

else

{

echo json\_encode(array('success' => false, 'message' => $this->lang->line('employees\_cannot\_be\_deleted')));

}

}

/\*

get the width for the add/edit form

\*/

function get\_form\_width()

{

return 650;

}

function data()

{

$index = isset($\_GET['order'][0]['column']) ? $\_GET['order'][0]['column'] : 1;

$dir = isset($\_GET['order'][0]['dir']) ? $\_GET['order'][0]['dir'] : "asc";

$order = array("index" => $index, "direction" => $dir);

$length = isset($\_GET['length']) ? $\_GET['length'] : 50;

$start = isset($\_GET['start']) ? $\_GET['start'] : 0;

$key = isset($\_GET['search']['value']) ? $\_GET['search']['value'] : "";

$people = $this->Employee->get\_all($length, $start, $key, $order);

$format\_result = array();

$width = 50;

foreach ($people->result() as $person)

{

$format\_result[] = array(

"<input type='checkbox' name='chk[]' id='person\_$person->person\_id' value='" . $person->person\_id . "'/>",

$person->person\_id,

$person->last\_name,

$person->first\_name,

$person->username,

$person->email,

$person->phone\_number,

anchor('employees/view/' . $person->person\_id, $this->lang->line('common\_edit'), array('class' => 'btn btn-success'))

);

}

$data = array(

"recordsTotal" => $this->Employee->count\_all(),

"recordsFiltered" => $this->Employee->count\_all(),

"data" => $format\_result

);

echo json\_encode($data);

exit;

}

function employee\_search()

{

$suggestions = $this->Employee->get\_employee\_search\_suggestions($this->input->get('query'), 30);

$data = $tmp = array();

foreach ($suggestions as $suggestion):

$t = explode("|", $suggestion);

$tmp = array("value" => $t[1], "data" => $t[0], "email" => $t[2]);

$data[] = $tmp;

endforeach;

echo json\_encode(array("suggestions" => $data));

exit;

}

}

?>

**-Model:**

<?php

class Employee extends Person {

/\*

Determines if a given person\_id is an employee

\*/

function exists($person\_id)

{

$this->db->from('employees');

$this->db->join('people', 'people.person\_id = employees.person\_id');

$this->db->where('employees.person\_id', $person\_id);

$query = $this->db->get();

return ($query->num\_rows() == 1);

}

/\*

Returns all the employees

\*/

function get\_all($limit = 10000, $offset = 0, $search = "", $order = array())

{

$user\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

$sorter = array("employees.person\_id", "employees.person\_id", "last\_name", "first\_name", "email", "phone\_number");

$this->db->from('employees');

$this->db->join('people', 'employees.person\_id=people.person\_id');

$this->db->where('deleted', 0);

if ($search !== "")

{

$this->db->where('first\_name LIKE ', '%' . $search . '%');

$this->db->or\_where('last\_name LIKE', '%' . $search . '%');

$this->db->or\_where('email LIKE', '%' . $search . '%');

$this->db->or\_where('phone\_number LIKE', '%' . $search . '%');

}

if (isset($\_GET['employee\_id']) && $\_GET['employee\_id'] > 0)

{

$user\_id = $\_GET['employee\_id'];

}

if ($user\_id !== '1')

{

$this->db->where('employees.added\_by', $user\_id);

}

if (count($order) > 0 && $order['index'] < count($sorter))

{

$this->db->order\_by($sorter[$order['index']], $order['direction']);

}

else

{

$this->db->order\_by("employees.person\_id", "desc");

}

$this->db->limit($limit);

$this->db->offset($offset);

return $this->db->get();

}

function count\_all()

{

$this->db->from('employees');

$this->db->where('deleted', 0);

return $this->db->count\_all\_results();

}

/\*

Gets information about a particular employee

\*/

function get\_info($employee\_id)

{

$this->db->from('employees');

$this->db->join('people', 'people.person\_id = employees.person\_id');

$this->db->where('employees.person\_id', $employee\_id);

$query = $this->db->get();

if ($query->num\_rows() == 1)

{

return $query->row();

}

else

{

//Get empty base parent object, as $employee\_id is NOT an employee

$person\_obj = parent::get\_info(-1);

//Get all the fields from employee table

$fields = $this->db->list\_fields('employees');

//append those fields to base parent object, we we have a complete empty object

foreach ($fields as $field)

{

$person\_obj->$field = '';

}

return $person\_obj;

}

}

/\*

Gets information about multiple employees

\*/

function get\_multiple\_info($employee\_ids)

{

$this->db->from('employees');

$this->db->join('people', 'people.person\_id = employees.person\_id');

$this->db->where\_in('employees.person\_id', $employee\_ids);

$this->db->order\_by("last\_name", "asc");

return $this->db->get();

}

/\*

Inserts or updates an employee

\*/

function save(&$person\_data, &$employee\_data, &$grants\_data, $employee\_id = false)

{

$success = false;

//Run these queries as a transaction, we want to make sure we do all or nothing

$this->db->trans\_start();

if (parent::save($person\_data, $employee\_id))

{

$employee\_data["added\_by"] = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

if (!$employee\_id or ! $this->exists($employee\_id))

{

$employee\_data['person\_id'] = $employee\_id = $person\_data['person\_id'];

$success = $this->db->insert('employees', $employee\_data);

}

else

{

$this->db->where('person\_id', $employee\_id);

$success = $this->db->update('employees', $employee\_data);

}

//We have either inserted or updated a new employee, now lets set permissions.

if ($success)

{

//First lets clear out any grants the employee currently has.

$success = $this->db->delete('grants', array('person\_id' => $employee\_id));

//Now insert the new grants

if ($success)

{

foreach ($grants\_data as $permission\_id)

{

$success = $this->db->insert('grants', array(

'permission\_id' => $permission\_id,

'person\_id' => $employee\_id));

}

}

}

}

$this->db->trans\_complete();

return $success;

}

/\*

Deletes one employee

\*/

function delete($employee\_id)

{

$success = false;

//Don't let employee delete their self

if ($employee\_id == $this->get\_logged\_in\_employee\_info()->person\_id)

return false;

//Run these queries as a transaction, we want to make sure we do all or nothing

$this->db->trans\_start();

//Delete permissions

if ($this->db->delete('grants', array('person\_id' => $employee\_id)))

{

$this->db->where('person\_id', $employee\_id);

$success = $this->db->update('employees', array('deleted' => 1));

}

$this->db->trans\_complete();

return $success;

}

/\*

Deletes a list of employees

\*/

function delete\_list($employee\_ids)

{

$success = false;

//Don't let employee delete their self

if (in\_array($this->get\_logged\_in\_employee\_info()->person\_id, $employee\_ids))

return false;

//Run these queries as a transaction, we want to make sure we do all or nothing

$this->db->trans\_start();

$this->db->where\_in('person\_id', $employee\_ids);

//Delete permissions

if ($this->db->delete('grants'))

{

//delete from employee table

$this->db->where\_in('person\_id', $employee\_ids);

$success = $this->db->update('employees', array('deleted' => 1));

}

$this->db->trans\_complete();

return $success;

}

/\*

Get search suggestions to find employees

\*/

function get\_search\_suggestions($search, $limit = 5)

{

$suggestions = array();

$this->db->from('employees');

$this->db->join('people', 'employees.person\_id=people.person\_id');

$this->db->where("(first\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

last\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

CONCAT(`first\_name`,' ',`last\_name`) LIKE '%" . $this->db->escape\_like\_str($search) . "%') and deleted=0");

$this->db->order\_by("last\_name", "asc");

$by\_name = $this->db->get();

foreach ($by\_name->result() as $row)

{

$suggestions[] = $row->first\_name . ' ' . $row->last\_name;

}

$this->db->from('employees');

$this->db->join('people', 'employees.person\_id=people.person\_id');

$this->db->where('deleted', 0);

$this->db->like("email", $search);

$this->db->order\_by("email", "asc");

$by\_email = $this->db->get();

foreach ($by\_email->result() as $row)

{

$suggestions[] = $row->email;

}

$this->db->from('employees');

$this->db->join('people', 'employees.person\_id=people.person\_id');

$this->db->where('deleted', 0);

$this->db->like("username", $search);

$this->db->order\_by("username", "asc");

$by\_username = $this->db->get();

foreach ($by\_username->result() as $row)

{

$suggestions[] = $row->username;

}

$this->db->from('employees');

$this->db->join('people', 'employees.person\_id=people.person\_id');

$this->db->where('deleted', 0);

$this->db->like("phone\_number", $search);

$this->db->order\_by("phone\_number", "asc");

$by\_phone = $this->db->get();

foreach ($by\_phone->result() as $row)

{

$suggestions[] = $row->phone\_number;

}

//only return $limit suggestions

if (count($suggestions > $limit))

{

$suggestions = array\_slice($suggestions, 0, $limit);

}

return $suggestions;

}

/\*

Preform a search on employees

\*/

function search($search)

{

$this->db->from('employees');

$this->db->join('people', 'employees.person\_id=people.person\_id');

$this->db->where("(first\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

last\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

email LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

phone\_number LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

username LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

CONCAT(`first\_name`,' ',`last\_name`) LIKE '%" . $this->db->escape\_like\_str($search) . "%') and deleted=0");

$this->db->order\_by("last\_name", "asc");

return $this->db->get();

}

/\*

Attempts to login employee and set session. Returns boolean based on outcome.

\*/

function login($username, $password)

{

$query = $this->db->get\_where('employees', array('username' => $username, 'password' => md5($password), 'deleted' => 0), 1);

if ($query->num\_rows() == 1)

{

$row = $query->row();

$this->session->set\_userdata('person\_id', $row->person\_id);

return true;

}

return false;

}

/\*

Logs out a user by destorying all session data and redirect to login

\*/

function logout()

{

$this->session->sess\_destroy();

redirect('login');

}

/\*

Determins if a employee is logged in

\*/

function is\_logged\_in()

{

return $this->session->userdata('person\_id') != false;

}

/\*

Gets information about the currently logged in employee.

\*/

function get\_logged\_in\_employee\_info()

{

if ($this->is\_logged\_in())

{

return $this->get\_info($this->session->userdata('person\_id'));

}

return false;

}

/\*

\* Determines whether the employee has access to at least one submodule

\*/

function has\_module\_grant($permission\_id, $person\_id)

{

$this->db->from('grants');

$this->db->like('permission\_id', $permission\_id, 'after');

$this->db->where('person\_id', $person\_id);

$result = $this->db->get();

$result\_count = $result->num\_rows();

if ($result\_count != 1)

{

return $result\_count != 0;

}

return $this->has\_subpermissions($permission\_id);

}

function has\_subpermissions($permission\_id)

{

$this->db->from('permissions');

$this->db->like('permission\_id', $permission\_id . '\_', 'after');

$result = $this->db->get();

return $result->num\_rows() == 0;

}

/\*

Determines whether the employee specified employee has access the specific module.

\*/

function has\_grant($permission\_id, $person\_id)

{

//if no module\_id is null, allow access

if ($permission\_id == null)

{

return true;

}

$query = $this->db->get\_where('grants', array('person\_id' => $person\_id, 'permission\_id' => $permission\_id), 1);

return ($query->num\_rows() == 1);

}

function get\_employee\_grants($person\_id)

{

$this->db->from('grants');

$this->db->where('person\_id', $person\_id);

return $this->db->get()->result\_array();

}

function hasLowerLevel()

{

$user = $this->Employee->get\_logged\_in\_employee\_info();

if ($user->person\_id === 1)

{

return true;

}

$this->db->where("role\_id", $user->role\_id);

$query = $this->db->get('roles');

if (isset($query->row()->low\_level) && $query->row()->low\_level !== "false")

return true;

return false;

}

function getLowerLevels()

{

$employee\_id = $this->session->userdata('person\_id');

$role\_id = $this->Employee->get\_logged\_in\_employee\_info()->role\_id;

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

$query = $this->db->get("roles");

$res = $query->row();

$low\_levels = false;

if (isset($res->low\_level) && is\_array(json\_decode($res->low\_level, TRUE)))

{

$low\_levels = json\_decode($res->low\_level, TRUE);

}

if ($low\_levels)

{

$this->db->where\_in("role\_id", $low\_levels);

$this->db->where("employees.added\_by", $employee\_id);

$this->db->join("employees", "employees.person\_id=people.person\_id");

$query = $this->db->get("people");

return $query->result();

}

return false;

}

/\*

Get search suggestions to find customers

\*/

function get\_employee\_search\_suggestions($search, $limit = 25)

{

$suggestions = array();

$user\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

$add\_where = '';

if ($user\_id > 1)

{

$add\_where = " and added\_by = " . $user\_id . " ";

}

$this->db->from('employees');

$this->db->join('people', 'employees.person\_id=people.person\_id');

$this->db->where("(first\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

last\_name LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

CONCAT(`first\_name`,' ',`last\_name`) LIKE '%" . $this->db->escape\_like\_str($search) . "%') and deleted=0

" . $add\_where . "

");

$this->db->order\_by("last\_name", "asc");

$by\_name = $this->db->get();

foreach ($by\_name->result() as $row)

{

$suggestions[] = $row->person\_id . '|' . $row->first\_name . ' ' . $row->last\_name . '|' . $row->email;

}

return $suggestions;

}

}

?>

**-View:**

<?php $this->load->view("partial/header"); ?>

<link href="css/plugins/iCheck/custom.css" rel="stylesheet">

<?php echo form\_open('employees/save/' . $person\_info->person\_id, array('id' => 'employee\_form', 'class' => 'form-horizontal')); ?>

<input type="hidden" id="person\_id" value="<?= $person\_info->person\_id ?>" />

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-content">

<h2><?php echo $this->lang->line('common\_list\_of') . ' ' . $this->lang->line('module\_' . $controller\_name); ?></h2>

<ol class="breadcrumb">

<li>

<a href="<?= site\_url(); ?>">Home</a>

</li>

<li>

<a href="<?= site\_url(strtolower($this->lang->line('module\_' . $controller\_name))); ?>"><?= ucwords($this->lang->line('module\_' . $controller\_name)); ?></a>

</li>

<li class="active">

<strong>Add</strong>

</li>

</ol>

</div>

</div>

</div>

</div>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-title">

<h5>

<?php echo $this->lang->line("employees\_basic\_information"); ?>

</h5>

<div class="inqbox-tools">

<a class="collapse-link">

<i class="fa fa-chevron-up"></i>

</a>

<a class="close-link">

<i class="fa fa-times"></i>

</a>

</div>

</div>

<div class="inqbox-content">

<div style="text-align: center">

<div id="required\_fields\_message"><?php echo $this->lang->line('common\_fields\_required\_message'); ?></div>

</div>

<?php $this->load->view("people/form\_basic\_info"); ?>

</div>

</div>

</div>

</div>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-title">

<h5>

<?php echo $this->lang->line("employees\_login\_info"); ?>

</h5>

<div class="inqbox-tools">

<a class="collapse-link">

<i class="fa fa-chevron-up"></i>

</a>

<a class="close-link">

<i class="fa fa-times"></i>

</a>

</div>

</div>

<div class="inqbox-content">

<br/>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('employees\_username') . ':', 'username'); ?></label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'username',

'id' => 'username',

'class' => 'form-control',

'value' => $person\_info->username

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<?php

$password\_label\_attributes = $person\_info->person\_id == "" ? array('class' => 'required') : array();

?>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('employees\_password') . ':', 'password', $password\_label\_attributes); ?></label>

<div class="col-sm-10">

<?php

echo form\_password(array(

'name' => 'password',

'id' => 'password',

'class' => 'form-control'

));

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('employees\_repeat\_password') . ':', 'repeat\_password', $password\_label\_attributes); ?></label>

<div class="col-sm-10">

<?php

echo form\_password(array(

'name' => 'repeat\_password',

'id' => 'repeat\_password',

'class' => 'form-control'

));

?>

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label" style="color:red">

Role

</label>

<div class="col-sm-10">

<select class="form-control" name="role\_id" id="role\_id">

<?php foreach ($roles as $role): ?>

<option value="<?= $role->role\_id; ?>" <?= ($role->role\_id === $person\_info->role\_id) ? "selected='selected'" : ""; ?>><?= $role->name; ?></option>

<?php endforeach; ?>

</select>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="form-group">

<div class="col-sm-4 col-sm-offset-2">

<button type="button" class="btn btn-default" data-dismiss="modal" id="btn-close"><?= $this->lang->line("common\_close"); ?></button>

<?php

echo form\_submit(array(

'name' => 'submit',

'id' => 'submit',

'value' => $this->lang->line('common\_submit'),

'class' => 'btn btn-primary')

);

?>

</div>

</div>

</div>

<?php

echo form\_close();

?>

<div id="feedback\_bar"></div>

<?php $this->load->view("partial/footer"); ?>

<script src="js/plugins/iCheck/icheck.min.js"></script>

<script src="<?php echo base\_url(); ?>js/people.js?v=<?= time(); ?>"></script>

<script>

$(document).ready(function () {

var person\_id = $("#person\_id").val();

$('.i-checks').iCheck({

checkboxClass: 'icheckbox\_square-green',

radioClass: 'iradio\_square-green'

});

var settings = {

submitHandler: function (form) {

$("#submit").prop("disabled", true);

$(form).ajaxSubmit({

success: function (response)

{

post\_person\_form\_submit(response);

$("#submit").prop("disabled", false);

},

dataType: 'json',

type: 'post'

});

},

errorPlacement: function (error, element) {

if (element.attr("name") === "grants[]")

{

error.appendTo('#error\_message\_box');

} else

{

error.insertAfter(element);

}

},

rules: {

first\_name: "required",

last\_name: "required",

username: {

required: true,

minlength: 5

},

password: {

required: true,

minlength: 8

},

repeat\_password: {

equalTo: "#password"

},

email: "email", "grants[]": {

required: function (element) {

var checked = false;

$("ul#permission\_list > li > input:checkbox").each(function () {

if ($(this).is(":checked"))

{

var has\_children = false;

$("ul > li > input:checkbox", $(this).parent()).each(function ()

{

has\_children = true;

checked |= $(this).is(":checked");

});

if (has\_children && !checked)

{

return false;

}

}

});

return !checked;

},

minlength: 1

}

},

messages: {

first\_name: "<?php echo $this->lang->line('common\_first\_name\_required'); ?>",

last\_name: "<?php echo $this->lang->line('common\_last\_name\_required'); ?>",

email: "<?php echo $this->lang->line('common\_email\_invalid\_format'); ?>",

username: {

required: "<?php echo $this->lang->line('employees\_username\_required'); ?>",

minlength: "<?php echo $this->lang->line('employees\_username\_minlength'); ?>"

},

password: {

required: "<?php echo $this->lang->line('employees\_password\_required'); ?>",

minlength: "<?php echo $this->lang->line('employees\_password\_minlength'); ?>"

},

repeat\_password: {

equalTo: "<?php echo $this->lang->line('employees\_password\_must\_match'); ?>"

},

"grants[]": "<?php echo $this->lang->line('employees\_grants\_required'); ?>"

}

};

if (person\_id !== "")

{

settings["rules"]["password"]["required"] = false;

}

// Validation

$('#employee\_form').validate(settings);

function post\_person\_form\_submit(response)

{

if (!response.success)

{

set\_feedback(response.message, 'error\_message', true);

} else

{

set\_feedback(response.message, 'success\_message', false);

}

$("#employee\_form").attr("action", "<?= site\_url(); ?>employees/save/" + response.person\_id);

}

});

</script>

**5.Loan**

**-Controller:**

<?php

require\_once ("secure\_area.php");

require\_once ("interfaces/idata\_controller.php");

class Loans extends Secure\_area implements iData\_controller {

function \_\_construct()

{

parent::\_\_construct('loans');

}

function index()

{

$data['controller\_name'] = strtolower(get\_class());

$data['form\_width'] = $this->get\_form\_width();

$data['count\_overdues'] = $this->\_count\_overdues();

$res = $this->Employee->getLowerLevels();

$data['staffs'] = $res;

$this->load->view('loans/manage', $data);

}

function search()

{

}

/\*

Gives search suggestions based on what is being searched for

\*/

function suggest()

{

}

function get\_row()

{

}

function view($loan\_id = -1)

{

$data['loan\_info'] = $this->Loan->get\_info($loan\_id);

$data['guarantee\_info'] = $this->Guarantee->get\_info($loan\_id);

$loan\_types = $this->Loan\_type->get\_multiple\_loan\_types();

print\_array($this->model->Loan->getRate());

$data['paymenterm'] = $this->model->Loan->getRate();

$tmp = array("" => $this->lang->line("common\_please\_select"));

foreach ($loan\_types as $loan\_type):

$tmp[$loan\_type->loan\_type\_id] = $loan\_type->name;

endforeach;

$data['loan\_types'] = $tmp;

$data['misc\_fees'] = array(

array("", "")

);

$misc\_fees = json\_decode($data['loan\_info']->misc\_fees, true);

if (is\_array($misc\_fees))

{

$tmp = array();

foreach ($misc\_fees as $fee => $charge):

$tmp[] = array($fee, $charge);

endforeach;

$data['misc\_fees'] = $tmp;

}

// payment scheds - start

$c\_payment\_sches = [

"term" => "",

"term\_period" => "",

"payment\_sched" => "",

"interest\_rate" => "",

"penalty" => "",

"payment\_breakdown" => [

"schedule" => [],

"balance" => [],

"interest" => [],

"amount" => []

]

];

$c\_payment\_scheds = trim($data["loan\_info"]->payment\_scheds) !== "" ? json\_decode($data["loan\_info"]->payment\_scheds, TRUE) : $c\_payment\_sches;

$data["c\_payment\_scheds"] = $c\_payment\_scheds;

// payment scheds - end

$attachments = $this->Loan->get\_attachments($loan\_id);

$file = array();

foreach ($attachments as $attachment)

{

$dFile = $this->\_get\_formatted\_file($attachment->attachment\_id, $attachment->filename, $attachment->descriptions);

$file[$dFile["id"]] = $dFile;

}

$data['attachments'] = $file;

$loan\_status = (isset($data['loan\_info']->loan\_status) && trim($data['loan\_info']->loan\_status) !== "") ? $this->lang->line("common\_" . strtolower($data['loan\_info']->loan\_status)) : $this->lang->line("common\_pending");

if ($data['loan\_info']->loan\_balance <= 0 && $loan\_id > -1)

{

$loan\_status = "paid";

}

$data['loan\_status'] = $loan\_status;

$employees = $this->Employee->get\_all()->result();

$emps = array();

foreach ($employees as $employee)

{

$emps[$employee->person\_id] = $employee->first\_name . " " . $employee->last\_name;

}

$data['employees'] = $emps;

$proof\_ids = json\_decode($data['guarantee\_info']->proof, TRUE);

$pimage\_ids = json\_decode($data['guarantee\_info']->images, TRUE);

$data["proofs"] = $this->\_get\_files($proof\_ids, $file);

$data["pimages"] = $this->\_get\_files($pimage\_ids, $file);

$this->load->view("loans/form", $data);

}

private function \_get\_files($ids, $file)

{

$tmp = array();

if (is\_array($ids))

{

foreach ($ids as $id):

$tmp[] = $file[$id];

endforeach;

}

return $tmp;

}

private function \_get\_formatted\_file($id, $filename, $desc)

{

$words = array("doc", "docx", "odt");

$xls = array("xls", "xlsx", "csv");

$tmp = explode(".", $filename);

$ext = $tmp[1];

if (in\_array(strtolower($ext), $words))

{

$tmp['icon'] = "images/word-filetype.jpg";

$tmp['filename'] = $filename;

$tmp['id'] = $id;

$tmp['descriptions'] = $desc;

}

else if (strtolower($ext) === "pdf")

{

$tmp['icon'] = "images/pdf-filetype.jpg";

$tmp['filename'] = $filename;

$tmp['id'] = $id;

$tmp['descriptions'] = $desc;

}

else if (in\_array(strtolower($ext), $xls))

{

$tmp['icon'] = "images/xls-filetype.jpg";

$tmp['filename'] = $filename;

$tmp['id'] = $id;

$tmp['descriptions'] = $desc;

}

else

{

$tmp['icon'] = "images/image-filetype.jpg";

$tmp['filename'] = $filename;

$tmp['id'] = $id;

$tmp['descriptions'] = $desc;

}

return $tmp;

}

function save($loan\_id = -1)

{

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

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

$misc\_fees = array();

for ($i = 0; $i < count($fees); $i++):

$misc\_fees[$fees[$i]] = $amounts[$i];

endfor;

// payment scheds - start

$c\_term = $this->input->post('term');

$c\_term\_period = $this->input->post('term\_period');

$c\_payment\_sched = $this->input->post('payment\_sched');

$c\_interest\_rate = $this->input->post('interest\_rate');

$c\_penalty = $this->input->post('penalty');

$c\_payment\_dates = $this->input->post("payment\_date");

$c\_payment\_balance = $this->input->post("payment\_balance");

$c\_payment\_interest = $this->input->post("payment\_interest");

$c\_payment\_amount = $this->input->post("payment\_amount");

$pay\_breakdowns = array();

$pay\_breakdowns["schedule"] = $c\_payment\_dates;

$pay\_breakdowns["balance"] = $c\_payment\_balance;

$pay\_breakdowns["interest"] = $c\_payment\_interest;

$pay\_breakdowns["amount"] = $c\_payment\_amount;

$payment\_scheds = [

"term" => $c\_term,

"term\_period" => $c\_term\_period,

"term\_period" => $c\_term\_period,

"payment\_sched" => $c\_payment\_sched,

"interest\_rate" => $c\_interest\_rate,

"penalty" => $c\_penalty,

"payment\_breakdown" => $pay\_breakdowns

];

// payment scheds - end

$loan\_data = array(

'account' => $this->input->post('account'),

'description' => $this->input->post('description'),

'loan\_type\_id' => $this->input->post('loan\_type\_id'),

'loan\_amount' => $this->input->post('amount'),

'customer\_id' => $this->input->post('customer'),

'loan\_applied\_date' => strtotime($this->input->post('apply\_date')),

'remarks' => $this->input->post('remarks'),

'loan\_agent\_id' => $this->input->post('agent'),

'loan\_approved\_by\_id' => $this->input->post('approver'),

'loan\_status' => ($this->input->post('approver') > 0) ? "approved" : $this->input->post("status"),

'misc\_fees' => json\_encode($misc\_fees),

'payment\_scheds' => json\_encode($payment\_scheds),

);

// check loan payment date

if ($loan\_data["loan\_type\_id"] > 0)

{

}

else

{

$loan\_data["loan\_payment\_date"] = strtotime($pay\_breakdowns["schedule"][0]);

}

$guarantee\_data = array(

'loan\_id' => $loan\_id,

'name' => $this->input->post("guarantee\_type"),

'type' => $this->input->post("guarantee\_name"),

'brand' => $this->input->post("guarantee\_brand"),

'make' => $this->input->post("guarantee\_make"),

'serial' => $this->input->post("guarantee\_serial"),

'proof' => json\_encode($this->input->post("proofs")),

'images' => json\_encode($this->input->post("images")),

'price' => $this->input->post("guarantee\_price"),

'observations' => $this->input->post("guarantee\_observations")

);

$has\_payment = false;

$paid\_amount = $this->input->post("paid\_amount");

// check if paid amount is greater than 0

if ($paid\_amount > 0)

{

$balance\_amount = $this->input->post("current\_balance") - $paid\_amount;

$loan\_data["loan\_balance"] = $balance\_amount;

$has\_payment = true;

}

if ($this->Loan->save($loan\_data, $loan\_id, $has\_payment))

{

//New Loan

if ($loan\_id == -1)

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('loans\_successful\_adding') . ' ' .

$loan\_data['account'], 'loan\_id' => $loan\_data['loan\_id']));

$loan\_id = $loan\_data['loan\_id'];

}

else //previous loan

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('loans\_successful\_updating') . ' ' .

$loan\_data['account'], 'loan\_id' => $loan\_id));

}

if ($has\_payment)

{

$payment\_data = array(

'account' => $this->input->post('account'),

'loan\_id' => $loan\_data['loan\_id'],

'customer\_id' => $this->input->post('customer'),

'paid\_amount' => $this->input->post('paid\_amount'),

'balance\_amount' => $balance\_amount,

'date\_paid' => strtotime($this->input->post('date\_paid')),

'remarks' => $this->input->post('remarks'),

'teller\_id' => $this->input->post('teller'),

);

$this->Payment->save($payment\_data, -1);

}

$this->Guarantee->save($guarantee\_data, $loan\_id);

}

else//failure

{

echo json\_encode(array('success' => false, 'message' => $this->lang->line('loans\_error\_adding\_updating') . ' ' .

$loan\_data['account'], 'loan\_id' => -1));

}

}

function delete()

{

$loans\_to\_delete = $this->input->post('ids');

if ($this->Loan->delete\_list($loans\_to\_delete))

{

echo json\_encode(array('success' => true, 'message' => $this->lang->line('loans\_successful\_deleted') . ' ' .

count($loans\_to\_delete) . ' ' . $this->lang->line('loans\_one\_or\_multiple')));

}

else

{

echo json\_encode(array('success' => false, 'message' => $this->lang->line('loans\_cannot\_be\_deleted')));

}

}

/\*

get the width for the add/edit form

\*/

function get\_form\_width()

{

return 360;

}

function data($status = "")

{

$sel\_user = $this->input->get("employee\_id");

$order = array("index" => $\_GET['order'][0]['column'], "direction" => $\_GET['order'][0]['dir']);

$loans = $this->Loan->get\_all($\_GET['length'], $\_GET['start'], $\_GET['search']['value'], $order, $status, $sel\_user);

$format\_result = array();

foreach ($loans->result() as $loan)

{

$loan\_status = $loan->loan\_status;

if ($loan->loan\_balance <= 0)

{

$loan\_status = "Paid";

}

$format\_result[] = array(

"<input type='checkbox' name='chk[]' id='loan\_$loan->loan\_id' value='" . $loan->loan\_id . "'/>",

$loan->loan\_id,

$loan->loan\_type,

$loan->account,

$loan->description,

to\_currency($loan->loan\_amount),

to\_currency($loan->loan\_balance),

ucwords($loan->customer\_name),

ucwords($loan->agent\_name),

ucwords($loan->approver\_name),

date("m/d/Y", $loan->loan\_applied\_date),

date("m/d/Y", $loan->loan\_payment\_date),

$this->lang->line("common\_" . strtolower($loan\_status)),

anchor('loans/view/' . $loan->loan\_id, $this->lang->line('common\_view'), array('class' => 'btn btn-success'))

);

}

$data = array(

"recordsTotal" => $this->Loan->count\_all(),

"recordsFiltered" => $this->Loan->count\_all(),

"data" => $format\_result

);

echo json\_encode($data);

exit;

}

function overdues()

{

$order = array("index" => $\_GET['order'][0]['column'], "direction" => $\_GET['order'][0]['dir']);

$loans = $this->Loan->get\_all($\_GET['length'], $\_GET['start'], $\_GET['search']['value'], $order, "overdue");

$format\_result = array();

foreach ($loans->result() as $loan)

{

$loan\_status = $loan->loan\_status;

if ($loan->loan\_balance <= 0)

{

$loan\_status = "Paid";

}

$format\_result[] = array(

"<input type='checkbox' name='chk[]' id='loan\_$loan->loan\_id' value='" . $loan->loan\_id . "'/>",

$loan->loan\_id,

$loan->loan\_type,

$loan->account,

$loan->description,

to\_currency($loan->loan\_amount),

to\_currency($loan->loan\_balance),

ucwords($loan->customer\_name),

ucwords($loan->agent\_name),

ucwords($loan->approver\_name),

date("m/d/Y", $loan->loan\_applied\_date),

date("m/d/Y", $loan->loan\_payment\_date),

$this->lang->line("common\_" . strtolower($loan\_status)),

anchor('loans/view/' . $loan->loan\_id, $this->lang->line('common\_view'), array('class' => 'modal\_link btn btn-success', 'data-toggle' => 'modal', 'data-target' => '#loan\_modal', "title" => "Update Loan"))

);

}

$data = array(

"recordsTotal" => $this->Loan->count\_all(),

"recordsFiltered" => $this->Loan->count\_all(),

"data" => $format\_result

);

echo json\_encode($data);

exit;

}

function fix\_breakdown($loan\_id)

{

$loan = $this->Loan->get\_info($loan\_id);

$loan\_type = $this->Loan\_type->get\_info($loan->loan\_type\_id);

$customer = $this->Customer->get\_info($loan->customer\_id);

$filename = "schedule" . time();

// As PDF creation takes a bit of memory, we're saving the created file in /downloads/reports/

$pdfFilePath = FCPATH . "/downloads/reports/$filename.pdf";

$data['company\_name'] = $this->config->item("company"); // company name

$data['company\_address'] = $this->config->item("address"); // company address

$data['phone'] = $this->config->item("phone"); // company address

$data['fax'] = $this->config->item("fax"); // company address

$data['email'] = $this->config->item("email"); // company address

$data['loan\_amount'] = to\_currency($loan->loan\_amount); // loan amount

$data['payable'] = to\_currency($loan->loan\_balance);

$data['rate'] = $loan\_type->percent\_charge1; // interest rate

$data['term'] = $loan\_type->term;

$data['loan'] = $loan;

$data['loan\_type'] = "Flexible";

$data['period\_type'] = $loan\_type->payment\_schedule;

$data['schedules'] = $this->Payment\_schedule->get\_schedules();

$data['misc\_fees'] = array();

$misc\_fees = json\_decode($loan->misc\_fees, true);

$total\_deductions = 0;

if (is\_array($misc\_fees))

{

$tmp = array();

foreach ($misc\_fees as $fee => $charge):

if (trim($charge) !== "")

{

$tmp[] = array($fee, to\_currency($charge));

}

$total\_deductions += $charge;

endforeach;

$data['misc\_fees'] = $tmp;

}

$data['customer\_name'] = ucwords($customer->first\_name . " " . $customer->last\_name);

$data['customer\_address'] = ucwords($customer->address\_1);

$data['total\_deductions'] = to\_currency($total\_deductions);

$data['net\_loan'] = to\_currency($loan->loan\_amount - $total\_deductions);

$data['repayment\_amount'] = to\_currency($payable);

$c\_payment\_sches = [

"term" => "",

"term\_period" => "",

"payment\_sched" => "",

"interest\_rate" => "",

"penalty" => "",

"payment\_breakdown" => [

"schedule" => [],

"balance" => [],

"interest" => [],

"amount" => []

]

];

$c\_payment\_scheds = trim($loan->payment\_scheds) !== "" ? json\_decode($loan->payment\_scheds, TRUE) : $c\_payment\_sches;

$data["c\_payment\_scheds"] = $c\_payment\_scheds;

//$data['breakdown\_data'] = $this->\_calculate\_breakdown($loan\_type->term, $period, $loan\_type->percent\_charge1, $payable, $loan->loan\_balance, $loan->loan\_payment\_date, $loan\_type->payment\_schedule);

ini\_set('memory\_limit', '64M'); // boost the memory limit if it's low <img src="https://davidsimpson.me/wp-includes/images/smilies/icon\_wink.gif" alt=";)" class="wp-smiley">

$html = $this->load->view('loans/payment\_schedule\_fix', $data, true); // render the view into HTML

$this->load->library('pdf');

$pdf = $this->pdf->load();

$pdf->SetFooter($\_SERVER['HTTP\_HOST'] . '|{PAGENO}|' . date(DATE\_RFC822)); // Add a footer for good measure <img src="https://davidsimpson.me/wp-includes/images/smilies/icon\_wink.gif" alt=";)" class="wp-smiley">

$pdf->WriteHTML($html); // write the HTML into the PDF

$pdf->Output($pdfFilePath, 'F'); // save to file because we can

redirect(base\_url("downloads/reports/$filename.pdf"));

}

function generate\_breakdown($loan\_id)

{

$loan = $this->Loan->get\_info($loan\_id);

$loan\_type = $this->Loan\_type->get\_info($loan->loan\_type\_id);

$customer = $this->Customer->get\_info($loan->customer\_id);

if ($loan\_type->term\_period\_type === "year")

{

$period = $this->\_get\_period($loan\_type->payment\_schedule);

}

else

{

$period = $this->\_get\_period($loan\_type->payment\_schedule, false);

}

$payable = $this->\_calculate\_mortgage($loan->loan\_balance, $loan\_type->percent\_charge1, $loan\_type->term, $period);

$filename = "schedule" . time();

// As PDF creation takes a bit of memory, we're saving the created file in /downloads/reports/

$pdfFilePath = FCPATH . "/downloads/reports/$filename.pdf";

$data['company\_name'] = $this->config->item("company"); // company name

$data['company\_address'] = $this->config->item("address"); // company address

$data['phone'] = $this->config->item("phone"); // company address

$data['fax'] = $this->config->item("fax"); // company address

$data['email'] = $this->config->item("email"); // company address

$data['loan\_amount'] = to\_currency($loan->loan\_amount); // loan amount

$data['payable'] = to\_currency($loan->loan\_balance);

$data['rate'] = $loan\_type->percent\_charge1; // interest rate

$data['term'] = $loan\_type->term;

$data['loan'] = $loan;

$data['loan\_type'] = $loan\_type;

$data['period\_type'] = $loan\_type->payment\_schedule;

$data['schedules'] = $this->Payment\_schedule->get\_schedules();

$data['misc\_fees'] = array();

$misc\_fees = json\_decode($loan->misc\_fees, true);

$total\_deductions = 0;

if (is\_array($misc\_fees))

{

$tmp = array();

foreach ($misc\_fees as $fee => $charge):

if (trim($charge) !== "")

{

$tmp[] = array($fee, to\_currency($charge));

}

$total\_deductions += $charge;

endforeach;

$data['misc\_fees'] = $tmp;

}

$data['customer\_name'] = ucwords($customer->first\_name . " " . $customer->last\_name);

$data['customer\_address'] = ucwords($customer->address\_1);

$data['total\_deductions'] = to\_currency($total\_deductions);

$data['net\_loan'] = to\_currency($loan->loan\_amount - $total\_deductions);

$data['total\_interest'] = $this->\_calculate\_total\_interest($loan->loan\_balance, $loan\_type->term, $payable, $period); // pass data to the view

if ($loan\_type->term\_period\_type === "year")

{

$tmp = $this->\_get\_repayment\_amount\_year\_term($loan\_type, $loan->loan\_amount);

}

else

{

$tmp = $this->\_get\_repayment\_amount\_month\_term($loan\_type, $loan->loan\_amount);

}

$data['repayment\_amount'] = to\_currency($payable);

$data['payment\_sched'] = strtoupper($tmp['payment\_sched']);

$data['apr'] = $tmp['apr'];

$data['breakdown\_data'] = $this->\_calculate\_breakdown($loan\_type->term, $period, $loan\_type->percent\_charge1, $payable, $loan->loan\_balance, $loan->loan\_payment\_date, $loan\_type->payment\_schedule);

ini\_set('memory\_limit', '64M'); // boost the memory limit if it's low <img src="https://davidsimpson.me/wp-includes/images/smilies/icon\_wink.gif" alt=";)" class="wp-smiley">

$html = $this->load->view('loans/payment\_schedule', $data, true); // render the view into HTML

$this->load->library('pdf');

$pdf = $this->pdf->load();

$pdf->SetFooter($\_SERVER['HTTP\_HOST'] . '|{PAGENO}|' . date(DATE\_RFC822)); // Add a footer for good measure <img src="https://davidsimpson.me/wp-includes/images/smilies/icon\_wink.gif" alt=";)" class="wp-smiley">

$pdf->WriteHTML($html); // write the HTML into the PDF

$pdf->Output($pdfFilePath, 'F'); // save to file because we can

redirect(base\_url("downloads/reports/$filename.pdf"));

}

function print\_disclosure($loan\_id)

{

$loan = $this->Loan->get\_info($loan\_id);

$loan\_type = $this->Loan\_type->get\_info($loan->loan\_type\_id);

$customer = $this->Customer->get\_info($loan->customer\_id);

if ($loan\_type->term\_period\_type === "year")

{

$period = $this->\_get\_period($loan\_type->payment\_schedule);

}

else

{

$period = $this->\_get\_period($loan\_type->payment\_schedule, false);

}

$payable = $this->\_calculate\_mortgage($loan->loan\_balance, $loan\_type->percent\_charge1, $loan\_type->term, $period);

$filename = "disclosure" . time();

// As PDF creation takes a bit of memory, we're saving the created file in /downloads/reports/

$pdfFilePath = FCPATH . "/downloads/reports/$filename.pdf";

$data['company\_name'] = $this->config->item("company"); // company name

$data['company\_address'] = $this->config->item("address"); // company address

$data['phone'] = $this->config->item("phone"); // company address

$data['fax'] = $this->config->item("fax"); // company address

$data['email'] = $this->config->item("email"); // company address

$data['loan\_amount'] = to\_currency($loan->loan\_amount); // loan amount

$data['payable'] = to\_currency($loan->loan\_balance);

$data['rate'] = $loan\_type->percent\_charge1; // interest rate

$data['term'] = $loan\_type->term;

$data['loan'] = $loan;

$data['loan\_type'] = $loan\_type;

$data['period\_type'] = $loan\_type->payment\_schedule;

$data['schedules'] = $this->Payment\_schedule->get\_schedules();

$data['misc\_fees'] = array();

$misc\_fees = json\_decode($loan->misc\_fees, true);

$total\_deductions = 0;

if (is\_array($misc\_fees))

{

$tmp = array();

foreach ($misc\_fees as $fee => $charge):

if (trim($charge) !== "")

{

$tmp[] = array($fee, to\_currency($charge));

}

$total\_deductions += $charge;

endforeach;

$data['misc\_fees'] = $tmp;

}

$data['customer\_name'] = ucwords($customer->first\_name . " " . $customer->last\_name);

$data['customer\_address'] = ucwords($customer->address\_1);

$data['total\_deductions'] = to\_currency($total\_deductions);

$data['net\_loan'] = to\_currency($loan->loan\_amount - $total\_deductions);

//$data['total\_interest'] = $this->\_calculate\_total\_interest($loan->loan\_balance, $loan\_type->term, $payable, $period); // pass data to the view

if ($loan\_type->term\_period\_type === "year")

{

$tmp = $this->\_get\_repayment\_amount\_year\_term($loan\_type, $loan->loan\_amount);

}

else

{

$tmp = $this->\_get\_repayment\_amount\_month\_term($loan\_type, $loan->loan\_amount);

}

$data['repayment\_amount'] = to\_currency($payable);

$data['payment\_sched'] = strtoupper($tmp['payment\_sched']);

$data['apr'] = $tmp['apr'];

//$data['breakdown\_data'] = $this->\_calculate\_breakdown($loan\_type->term, $period, $loan\_type->percent\_charge1, $payable, $loan->loan\_balance, $loan->loan\_payment\_date);

//if (file\_exists($pdfFilePath) == FALSE)

//{

ini\_set('memory\_limit', '64M'); // boost the memory limit if it's low <img src="https://davidsimpson.me/wp-includes/images/smilies/icon\_wink.gif" alt=";)" class="wp-smiley">

$html = $this->load->view('loans/pdf\_report', $data, true); // render the view into HTML

$this->load->library('pdf');

$pdf = $this->pdf->load();

$pdf->SetFooter($\_SERVER['HTTP\_HOST'] . '|{PAGENO}|' . date(DATE\_RFC822)); // Add a footer for good measure <img src="https://davidsimpson.me/wp-includes/images/smilies/icon\_wink.gif" alt=";)" class="wp-smiley">

$pdf->WriteHTML($html); // write the HTML into the PDF

$pdf->Output($pdfFilePath, 'F'); // save to file because we can

//}

redirect(base\_url("downloads/reports/$filename.pdf"));

}

function fix\_disclosure($loan\_id)

{

$loan = $this->Loan->get\_info($loan\_id);

$loan\_type = $this->Loan\_type->get\_info($loan->loan\_type\_id);

$customer = $this->Customer->get\_info($loan->customer\_id);

$filename = "disclosure" . time();

// As PDF creation takes a bit of memory, we're saving the created file in /downloads/reports/

$pdfFilePath = FCPATH . "/downloads/reports/$filename.pdf";

$data['company\_name'] = $this->config->item("company"); // company name

$data['company\_address'] = $this->config->item("address"); // company address

$data['phone'] = $this->config->item("phone"); // company address

$data['fax'] = $this->config->item("fax"); // company address

$data['email'] = $this->config->item("email"); // company address

$data['loan\_amount'] = to\_currency($loan->loan\_amount); // loan amount

$data['payable'] = to\_currency($loan->loan\_balance);

$data['rate'] = $loan\_type->percent\_charge1; // interest rate

$data['term'] = $loan\_type->term;

$data['loan'] = $loan;

$data['loan\_type'] = $loan\_type;

$data['period\_type'] = $loan\_type->payment\_schedule;

$data['schedules'] = $this->Payment\_schedule->get\_schedules();

$data['misc\_fees'] = array();

$misc\_fees = json\_decode($loan->misc\_fees, true);

$total\_deductions = 0;

if (is\_array($misc\_fees))

{

$tmp = array();

foreach ($misc\_fees as $fee => $charge):

if (trim($charge) !== "")

{

$tmp[] = array($fee, to\_currency($charge));

}

$total\_deductions += $charge;

endforeach;

$data['misc\_fees'] = $tmp;

}

$data['customer\_name'] = ucwords($customer->first\_name . " " . $customer->last\_name);

$data['customer\_address'] = ucwords($customer->address\_1);

$data['total\_deductions'] = to\_currency($total\_deductions);

$data['net\_loan'] = to\_currency($loan->loan\_amount - $total\_deductions);

$data['repayment\_amount'] = to\_currency($payable);

$c\_payment\_sches = [

"term" => "",

"term\_period" => "",

"payment\_sched" => "",

"interest\_rate" => "",

"penalty" => "",

"payment\_breakdown" => [

"schedule" => [],

"balance" => [],

"interest" => [],

"amount" => []

]

];

$c\_payment\_scheds = trim($loan->payment\_scheds) !== "" ? json\_decode($loan->payment\_scheds, TRUE) : $c\_payment\_sches;

$data["c\_payment\_scheds"] = $c\_payment\_scheds;

ini\_set('memory\_limit', '64M'); // boost the memory limit if it's low <img src="https://davidsimpson.me/wp-includes/images/smilies/icon\_wink.gif" alt=";)" class="wp-smiley">

$html = $this->load->view('loans/pdf\_report\_fix', $data, true); // render the view into HTML

$this->load->library('pdf');

$pdf = $this->pdf->load();

$pdf->SetFooter($\_SERVER['HTTP\_HOST'] . '|{PAGENO}|' . date(DATE\_RFC822)); // Add a footer for good measure <img src="https://davidsimpson.me/wp-includes/images/smilies/icon\_wink.gif" alt=";)" class="wp-smiley">

$pdf->WriteHTML($html); // write the HTML into the PDF

$pdf->Output($pdfFilePath, 'F'); // save to file because we can

redirect(base\_url("downloads/reports/$filename.pdf"));

}

private function \_get\_repayment\_amount\_year\_term($loan\_type, $loan\_balance\_amount)

{

// get the term of payment

$term = $loan\_type->term; // 1yr

// period\_type1 means recurrence of interest

// how to get the APR (Annual Percentage Rate) if

// im going to give an interest rate of 3% every 3 weeks

// of course im going get how many time 3 weeks in a year

// 52 weeks / 3 weeks = 17 weeks then multiply it by 3%

// the answer is the APR, which is 51% in year.

// interest rate is computed by APR / number times in a year

switch ($loan\_type->period\_type1)

{

case "Week": // 52 weeks in 1yr

$period = 52 \* $term;

$apr = (52 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $apr / 52;

break;

case "Month": // 12 months in 1yr

$period = 12 \* $term;

$apr = (12 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $apr / 12;

break;

case "Year": // 1yr in 1yr

$period = 1 \* $term;

$apr = (1 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $apr / 1;

break;

}

switch ($loan\_type->payment\_schedule)

{

case "weekly": // 52 weeks in 1yr

$factor = 52 \* $term;

$payment\_sched = "Weekly payment";

break;

case "monthly": // 12 months in 1yr

$factor = 12 \* $term;

$payment\_sched = "Monthly payment";

break;

case "yearly": // 1yr in 1yr

$factor = 1 \* $term;

$payment\_sched = "Yearly payment";

break;

case "daily":

$factor = 365 \* $term;

$payment\_sched = "Daily payment";

break;

}

if ($apr > 0)

{

$rate\_per\_payment = 1 / (1 + ($interest\_rate / 100) );

$payment = $loan\_balance\_amount \* ( (1 - $rate\_per\_payment) / ( $rate\_per\_payment - (pow($rate\_per\_payment, $factor)) ) );

return array("repayment\_amount" => $payment, "apr" => $apr, "payment\_sched" => $payment\_sched);

}

else

{

$payment = $loan\_balance\_amount / $factor;

return array("repayment\_amount" => $payment, "apr" => $apr, "payment\_sched" => $payment\_sched);

}

}

private function \_get\_repayment\_amount\_month\_term($loan\_type, $loan\_balance\_amount)

{

// get the term of payment

$term = $loan\_type->term; // 1month

// period\_type1 means recurrence of interest

// how to get the APR (Annual Percentage Rate) if

// im going to give an interest rate of 3% every 3 weeks

// of course im going get how many time 3 weeks in a year

// 52 weeks / 3 weeks = 17 weeks then multiply it by 3%

// the answer is the APR, which is 51% in year.

// interest rate is computed by APR / number times in a year

/\* switch ($loan\_type->period\_type1)

{

case "Week": // 52 weeks in 1yr

$period = 52 \* $term;

$amr = (52 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $amr / 52;

break;

case "Month": // 12 months in 1yr

$period = 12 \* $term;

$amr = (12 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $amr / 12;

break;

case "Year": // 1yr in 1yr

$period = 1 \* $term;

$amr = (1 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $amr / 1;

break;

} \*/

switch ($loan\_type->payment\_schedule)

{

case "weekly": // 4 weeks in 1month

$factor = 4 \* $term;

$payment\_sched = "Weekly payment";

break;

case "biweekly": // 4 weeks in 1month

$factor = 2 \* $term;

$payment\_sched = "Bi-Weekly payment";

break;

case "monthly": // 1 month in 1month

$factor = 1 \* $term;

$payment\_sched = "Monthly payment";

break;

case "daily":

$factor = 30 \* $term;

$payment\_sched = "Daily payment";

break;

}

$interest\_rate = $loan\_type->percent\_charge1;

if ($interest\_rate > 0)

{

$rate\_per\_payment = 1 / (1 + ($interest\_rate / 100) );

$repayment = $loan\_balance\_amount \* ( (1 - $rate\_per\_payment) / ( $rate\_per\_payment - (pow($rate\_per\_payment, $factor)) ) );

return array("repayment\_amount" => $repayment, "apr" => $interest\_rate, "payment\_sched" => $payment\_sched);

}

else

{

$payment = $loan\_balance\_amount / $factor;

return array("repayment\_amount" => $payment, "apr" => $amr, "payment\_sched" => $payment\_sched);

}

}

private function \_calculate\_breakdown($term, $period, $rate, $pay, $balance, $payment\_date, $payment\_schedule)

{

$data = array();

for ($i = 0; $i < ($term \* $period); $i++)

{

$tmp = (($pay) - ($balance \* ($rate / 100 / $period)));

$diff = round($tmp, 2);

$int = round(($balance \* $rate / 100 / $period), 2);

$princ = $balance - $diff;

$balance = round($balance, 0);

$data[$i]['month'] = date("M d, Y", $payment\_date);

switch ($payment\_schedule)

{

case "weekly":

$payment\_date = strtotime("+7 day", $payment\_date);

break;

case "biweekly":

$payment\_date = strtotime("+14 day", $payment\_date);

break;

case "monthly":

$payment\_date = strtotime("+1 month", $payment\_date);

break;

case "bimonthly":

$payment\_date = strtotime("+2 month", $payment\_date);

break;

case "daily":

$payment\_date = strtotime("+1 day", $payment\_date);

break;

}

$data[$i]['balance'] = to\_currency($balance);

$data[$i]['interest'] = $int;

$data[$i]['pay'] = to\_currency($pay);

$balance = $princ;

}

return $data;

}

private function \_get\_period($period\_type, $is\_yearly = true)

{

if ($is\_yearly)

{

// 52 - Weekly

// 26 - Biweekly

// 12 - Monthly

// 6 - Bimonthly

// $period = 12;

$period = 12;

switch ($period\_type)

{

case "weekly":

$period = 52;

break;

case "biweekly":

$period = 26;

break;

case "monthly":

$period = 12;

break;

case "bimonthly":

$period = 6;

break;

}

}

else

{

// 4 - Weekly

// $period = 12;

$period = 4;

switch ($period\_type)

{

case "weekly":

$period = 4;

break;

case "biweekly":

$period = 2;

break;

case "daily":

$period = 30;

break;

case "monthly":

$period = 1;

break;

}

}

return $period;

}

private function \_calculate\_mortgage($balance, $rate, $term, $period)

{

$N = (int) $term \* (int) $period;

$I = ((float) $rate / 100) / (int) $period;

$v = pow((1 + $I), $N);

$t = ($I \* $v) / ($v - 1);

$result = $balance \* $t;

return $result;

}

private function \_calculate\_total\_interest($balance, $term, $pay, $period)

{

return (($term \* $pay \* $period) - $balance);

}

private function \_count\_overdues()

{

return $this->Loan->count\_overdues();

}

function customer\_search()

{

$suggestions = $this->Customer->get\_customer\_search\_suggestions($this->input->get('query'), 30);

$data = $tmp = array();

foreach ($suggestions as $suggestion):

$t = explode("|", $suggestion);

$tmp = array("value" => $t[1], "data" => $t[0]);

$data[] = $tmp;

endforeach;

echo json\_encode(array("suggestions" => $data));

exit;

}

function select\_customer()

{

$customer\_id = $this->input->post("customer");

$this->sale\_lib->set\_customer($customer\_id);

$this->\_reload();

}

function upload()

{

$directory = FCPATH . 'uploads/loan-' . $\_REQUEST["loan\_id"] . "/";

$this->load->library('uploader');

$data = $this->uploader->upload($directory);

$this->Loan->save\_attachments($data['params']['loan\_id'], $data);

$file = $this->\_get\_formatted\_file($data['attachment\_id'], $data['filename'], "");

$file['loan\_id'] = $data['params']['loan\_id'];

$file['id'] = $data["attachment\_id"];

echo json\_encode($file);

exit;

}

function remove\_file()

{

$file\_id = $this->input->post("file\_id");

echo json\_encode(array("status" => $this->Loan->remove\_file($file\_id)));

exit;

}

function attach\_desc()

{

$id = $this->input->post("attach\_id");

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

$this->Loan->save\_attach\_desc($id, $desc);

echo json\_encode(array("success" => TRUE));

exit;

}

function attachments($loan\_id, $select\_type)

{

$data['loan\_info'] = $this->Loan->get\_info($loan\_id);

$attachments = $this->Loan->get\_attachments($loan\_id);

$file = array();

foreach ($attachments as $attachment)

{

$file[] = $this->\_get\_formatted\_file($attachment->attachment\_id, $attachment->filename, $attachment->descriptions);

}

$data["select\_type"] = $select\_type;

$data['attachments'] = $file;

$this->load->view("loans/attachments", $data);

}

}

?>

**-Model:**

<?php

class Loan extends CI\_Model {

/\*

Determines if a given loan\_id is a loan

\*/

function exists($loan\_id)

{

$this->db->from('loans');

$this->db->where('loan\_id', $loan\_id);

$query = $this->db->get();

return ($query->num\_rows() == 1);

}

function get\_all($limit = 10000, $offset = 0, $search = "", $order = array(), $status = "", $sel\_user = false)

{

$employee\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

$sorter = array(

"loan\_id",

"loan\_id",

"loan\_types.name",

"account",

"description",

"loan\_amount",

"loan\_balance",

"customer.first\_name",

"agent.first\_name",

"approver.first\_name",

"loan\_applied\_date",

"loan\_payment\_date",

"loan\_status",

);

$select = "loans.\*, CONCAT(customer.first\_name, ' ', customer.last\_name) as customer\_name,

CONCAT(agent.first\_name, ' ',agent.last\_name) as agent\_name,

CONCAT(approver.first\_name, ' ', approver.last\_name) as approver\_name,

IF (

c19\_loans.loan\_type\_id = 0,

'Flexible',

loan\_types.name

) AS loan\_type";

$this->db->select($select, FALSE);

$this->db->from('loans');

$this->db->join('people as customer', 'customer.person\_id = loans.customer\_id', 'LEFT');

$this->db->join('people as agent', 'agent.person\_id = loans.loan\_agent\_id', 'LEFT');

$this->db->join('people as approver', 'approver.person\_id = loans.loan\_approved\_by\_id', 'LEFT');

$this->db->join('loan\_types', 'loan\_types.loan\_type\_id = loans.loan\_type\_id', 'LEFT');

$employee\_id = ($sel\_user) ? $sel\_user : $employee\_id;

if ($employee\_id > 1)

{

$this->db->where("loan\_agent\_id", $employee\_id);

}

if ($search !== "")

{

// hard coded the kpos\_loans table here might bug in the future

$this->db->where("(

account LIKE '%" . $search . "%' OR

kpos\_loans.description LIKE '%" . $search . "%' OR

customer.first\_name LIKE '%" . $search . "%' OR

customer.last\_name LIKE '%" . $search . "%' OR

CONCAT(customer.first\_name,' ', customer.last\_name) LIKE '%" . $search . "%' OR

kpos\_loan\_types.name LIKE '%" . $search . "%' OR

agent.first\_name LIKE '%" . $search . "%' OR

agent.last\_name LIKE '%" . $search . "%' OR

CONCAT(agent.first\_name, ' ', agent.last\_name) LIKE '%" . $search . "%'

)");

}

if (count($order) > 0 && $order['index'] < count($sorter))

{

$this->db->order\_by($sorter[$order['index']], $order['direction']);

}

else

{

$this->db->order\_by("loan\_id", "desc");

}

$this->db->where('delete\_flag', 0);

if ($status !== "")

{

if ($status === "paid")

{

$this->db->where("loan\_balance", 0);

}

else if ($status === "unpaid")

{

$this->db->where("loan\_balance >", 0);

}

else if ($status === "overdue")

{

$this->db->where("loan\_payment\_date < UNIX\_TIMESTAMP()");

}

}

$this->db->limit($limit);

$this->db->offset($offset);

return $this->db->get();

}

function count\_all()

{

$this->db->from('loans');

$this->db->where("delete\_flag", 0);

return $this->db->count\_all\_results();

}

function getRate()

{

$this->db->select('\*');

$this->db->from('paymenterm');

$query = $this->db->get();

if ($query->num\_rows() > 0)

{

return $query->result();

}

}

function count\_overdues()

{

$employee\_id = $this->Employee->get\_logged\_in\_employee\_info()->person\_id;

$this->db->where("loan\_payment\_date < UNIX\_TIMESTAMP()");

$this->db->from('loans');

$this->db->where("delete\_flag", 0);

if ($employee\_id > 1)

{

$this->db->where("loan\_agent\_id", $employee\_id);

}

return $this->db->count\_all\_results();

}

/\*

Gets information about a particular loan

\*/

function get\_info($loan\_id)

{

$select = "loans.\*, CONCAT(customer.first\_name, ' ', customer.last\_name) as customer\_name,

CONCAT(agent.first\_name, ' ',agent.last\_name) as agent\_name,

CONCAT(approver.first\_name, ' ', approver.last\_name) as approver\_name";

$this->db->select($select, FALSE);

$this->db->from('loans');

$this->db->join('people as customer', 'customer.person\_id = loans.customer\_id', 'LEFT');

$this->db->join('people as agent', 'agent.person\_id = loans.loan\_agent\_id', 'LEFT');

$this->db->join('people as approver', 'approver.person\_id = loans.loan\_approved\_by\_id', 'LEFT');

$this->db->where('loan\_id', $loan\_id);

$query = $this->db->get();

if ($query->num\_rows() == 1)

{

return $query->row();

}

else

{

//Get empty base parent object, as $loan\_id is NOT a loan

$loan\_obj = new stdClass();

//Get all the fields from items table

$fields = $this->db->list\_fields('loans');

foreach ($fields as $field)

{

$loan\_obj->$field = '';

}

$loan\_obj->loan\_id = -1;

$loan\_obj->customer\_name = '';

$loan\_obj->loan\_status = 'pending';

return $loan\_obj;

}

}

/\*

Gets information about multiple loans

\*/

function get\_multiple\_info($loans\_ids)

{

$this->db->from('loans');

$this->db->where\_in('item\_kit\_id', $loans\_ids);

$this->db->order\_by("account", "asc");

return $this->db->get();

}

/\*

Inserts or updates a loan

\*/

function save(&$loan\_data, $loan\_id = false, $has\_payment = false)

{

if (!$has\_payment)

{

$loan\_data['loan\_balance'] = $this->\_get\_loan\_balance($loan\_data);

}

if ($loan\_data["loan\_type\_id"] > 0)

{

$loan\_data['loan\_payment\_date'] = $this->\_get\_loan\_payment\_date($loan\_data);

}

if (!$loan\_id or ! $this->exists($loan\_id))

{

if ($this->db->insert('loans', $loan\_data))

{

$loan\_data['loan\_id'] = $this->db->insert\_id();

$this->move\_attachments($loan\_data);

return true;

}

return false;

}

$this->db->where('loan\_id', $loan\_id);

return $this->db->update('loans', $loan\_data);

}

private function \_get\_loan\_balance($loan\_data)

{

$loan\_type = $this->Loan\_type->get\_info($loan\_data['loan\_type\_id']);

$num\_weeks = ($loan\_type->term\_period\_type === "year") ? 52 : 4;

$num\_months = ($loan\_type->term\_period\_type === "year") ? 12 : 1;

if ($loan\_type->term\_period\_type === "year")

{

switch ($loan\_type->period\_type1)

{

case "Week": // 52 weeks in 1yr

$apr = (52 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $apr / $num\_weeks;

break;

case "Month": // 12 months in 1yr

$apr = (12 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $apr / $num\_months;

break;

case "Year": // 1yr in 1yr

$apr = (1 / $loan\_type->period\_charge1) \* $loan\_type->percent\_charge1;

$interest\_rate = $apr / 1;

break;

}

}

else if($loan\_type->term\_period\_type === "month")

{

$interest\_rate = $loan\_type->percent\_charge1;

}

else

{

$interest\_rate = 0;

}

return $loan\_data['loan\_amount'] + ($loan\_data['loan\_amount'] \* ($interest\_rate / 100) );

}

private function \_get\_loan\_payment\_date($loan\_data)

{

// 52 - Weekly

// 26 - Biweekly

// 12 - Monthly

// 6 - Bimonthly

// 365- Daily

$time = 0;

$loan\_type = $this->Loan\_type->get\_info($loan\_data['loan\_type\_id']);

switch ($loan\_type->payment\_schedule)

{

case "weekly":

$time = strtotime("+7 day", $loan\_data['loan\_applied\_date']);

break;

case "biweekly":

$time = strtotime("+14 day", $loan\_data['loan\_applied\_date']);

break;

case "monthly":

$time = strtotime("+1 month", $loan\_data['loan\_applied\_date']);

break;

case "bimonthly":

$time = strtotime("+2 month", $loan\_data['loan\_applied\_date']);

break;

case "daily":

$time = strtotime("+1 day", $loan\_data['loan\_applied\_date']);

break;

}

return $time;

}

/\*

\* Move attachment to the right location

\*/

function move\_attachments($loan\_data)

{

$linker = $this->session->userdata('linker');

$this->db->from('attachments');

$this->db->where('session\_id', $linker);

$query = $this->db->get();

$this->db->where('session\_id', $linker);

$this->db->update('attachments', array("loan\_id" => $loan\_data['loan\_id']));

$attachments = $query->result();

foreach ($attachments as $attachment)

{

$tmp\_dir = FCPATH . "uploads/loan--1/";

$user\_dir = FCPATH . "uploads/loan-" . $loan\_data['loan\_id'] . "/";

if (!file\_exists($user\_dir))

{

// temporary set to full access

@mkdir($user\_dir);

}

$target\_dist = $user\_dir . $attachment->filename;

if (file\_exists($tmp\_dir . $attachment->filename))

{

copy($tmp\_dir . $attachment->filename, $target\_dist);

unlink($tmp\_dir . $attachment->filename);

}

}

}

/\*

Deletes one loan

\*/

function delete($loan\_id)

{

$this->db->where('loan\_id', $loan\_id);

return $this->db->delete('loans', array('delete\_flag' => 1));

}

/\*

Deletes a list of loans

\*/

function delete\_list($loan\_ids)

{

$this->db->where\_in('loan\_id', $loan\_ids);

return $this->db->update('loans', array("delete\_flag" => 1));

}

/\*

Get search suggestions to find loans

\*/

function get\_search\_suggestions($search, $limit = 25)

{

$suggestions = array();

$this->db->from('loans');

$this->db->like('account', $search);

$this->db->order\_by("account", "asc");

$by\_name = $this->db->get();

foreach ($by\_name->result() as $row)

{

$suggestions[] = $row->account;

}

//only return $limit suggestions

if (count($suggestions > $limit))

{

$suggestions = array\_slice($suggestions, 0, $limit);

}

return $suggestions;

}

function get\_loan\_search\_suggestions($search, $limit = 25)

{

$suggestions = array();

$this->db->from('loans');

$this->db->like('account', $search);

$this->db->order\_by("account", "asc");

$by\_name = $this->db->get();

foreach ($by\_name->result() as $row)

{

$suggestions[] = 'LOAN ' . $row->item\_kit\_id . '|' . $row->name;

}

//only return $limit suggestions

if (count($suggestions > $limit))

{

$suggestions = array\_slice($suggestions, 0, $limit);

}

return $suggestions;

}

/\*

Preform a search on loans

\*/

function search($search)

{

$this->db->from('loans');

$this->db->where("account LIKE '%" . $this->db->escape\_like\_str($search) . "%' or

description LIKE '%" . $this->db->escape\_like\_str($search) . "%'");

$this->db->order\_by("account", "asc");

return $this->db->get();

}

function \_get\_count\_payments($loan\_id)

{

$this->db->where("loan\_id", $loan\_id);

$cnt = $this->db->count\_all\_results("loan\_payments");

return $cnt;

}

/\*

\* Perform update/insert balance

\*/

function update\_balance($amount, $loan\_id)

{

$loan = $this->get\_info($loan\_id);

$loan\_data['loan\_type\_id'] = $loan->loan\_type\_id;

$loan\_data['loan\_applied\_date'] = time();

if ($loan->loan\_type\_id > 0)

{

$due\_date = $this->\_get\_loan\_payment\_date($loan\_data);

}

else

{

$c\_payment\_scheds = json\_decode($loan->payment\_scheds, TRUE);

$count\_payments = $this->\_get\_count\_payments($loan->loan\_id);

$due\_date = time();

if (isset($c\_payment\_scheds["payment\_breakdown"]["schedule"][1]))

{

$due\_date = strtotime($c\_payment\_scheds["payment\_breakdown"]["schedule"][1]);

}

if ( isset($c\_payment\_scheds["payment\_breakdown"]["schedule"][$count\_payments + 1]) )

{

$due\_date = strtotime($c\_payment\_scheds["payment\_breakdown"]["schedule"][$count\_payments + 1]);

}

}

$this->db->trans\_start();

$new\_balance = $loan->loan\_balance - $amount;

$loan\_data['loan\_amount'] = $new\_balance;

$new\_balance = $this->\_get\_loan\_balance($loan\_data);

$data = array(

'loan\_balance' => $new\_balance,

'loan\_payment\_date' => $due\_date

);

$this->db->where('loan\_id', $loan\_id);

$this->db->update('loans', $data);

$this->db->trans\_complete();

}

function save\_attachments($loan\_id, &$data)

{

if ($loan\_id > 0)

{

if ($this->db->insert('attachments', array("filename" => $data['filename'], "loan\_id" => $loan\_id)))

{

$data['attachment\_id'] = $this->db->insert\_id();

return true;

}

}

$session\_id = $data['params']['linker'];

$this->load->library('session');

$this->session->set\_userdata(array("linker" => $session\_id));

if ($this->db->insert('attachments', array("filename" => $data['filename'], "session\_id" => $session\_id)))

{

$data['attachment\_id'] = $this->db->insert\_id();

return true;

}

}

function save\_attach\_desc($id, $desc)

{

$this->db->where('attachment\_id', $id);

return $this->db->update('attachments', array("descriptions" => $desc));

}

function get\_attachments($loan\_id)

{

$this->db->from('attachments');

$this->db->where('loan\_id', $loan\_id);

$query = $this->db->get();

return $query->result();

}

function remove\_file($file\_id)

{

$this->db->from('attachments');

$this->db->where('attachment\_id', $file\_id);

$query = $this->db->get();

$res = $query->row();

$user\_dir = FCPATH . "uploads/loan--1/";

if ($res->loan\_id > 0)

{

$user\_dir = FCPATH . "uploads/loan-" . $res->loan\_id . "/";

}

if (file\_exists($user\_dir . $res->filename))

{

unlink($user\_dir . $res->filename);

}

return $this->db->delete('attachments', array('attachment\_id' => $file\_id));

}

function get\_total\_loans()

{

$this->db->select("SUM(loan\_amount) as total\_loans");

$this->db->from("loans");

$this->db->where("loan\_status", "pending");

$this->db->where("delete\_flag", "0");

$query = $this->db->get();

$res = $query->row();

return to\_currency($res->total\_loans, TRUE, 0);

}

}

?>

**-View:**

<?php $this->load->view("partial/header"); ?>

<link href="css/plugins/datapicker/datepicker3.css" rel="stylesheet">

<?php echo form\_open('loans/save/' . $loan\_info->loan\_id, array('id' => 'loan\_form', 'class' => 'form-horizontal')); ?>

<style>

#drop-target {

border: 10px dashed #999;

text-align: center;

color: #999;

font-size: 20px;

width: 600px;

height: 300px;

line-height: 300px;

cursor: pointer;

}

#drop-target.dragover {

background: rgba(255, 255, 255, 0.4);

border-color: green;

}

</style>

<input type="hidden" id="loan\_id" name="loan\_id" value="<?= $loan\_info->loan\_id; ?>" />

<input type="hidden" id="controller" value="<?= strtolower($this->lang->line('module\_' . $controller\_name)); ?>" />

<input type="hidden" id="linker" value="<?= random\_string('alnum', 16); ?>" />

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-content">

<h2><?php echo $this->lang->line('common\_list\_of') . ' ' . $this->lang->line('module\_' . $controller\_name); ?></h2>

<ol class="breadcrumb">

<li>

<a href="<?= site\_url(); ?>">Home</a>

</li>

<li>

<a href="<?= site\_url(strtolower($this->lang->line('module\_' . $controller\_name))); ?>"><?= ucwords($this->lang->line('module\_' . $controller\_name)); ?></a>

</li>

<li class="active">

<strong>Add</strong>

</li>

</ol>

</div>

</div>

</div>

</div>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-title">

<h5>

<?php echo $this->lang->line("loans\_info"); ?>

</h5>

<div class="inqbox-tools">

<a class="collapse-link">

<i class="fa fa-chevron-up"></i>

</a>

<a class="dropdown-toggle" data-toggle="dropdown" href="#">

<i class="fa fa-wrench"></i>

</a>

<ul class="dropdown-menu dropdown-user">

<li><a href="#">Config option 1</a>

</li>

<li><a href="#">Config option 2</a>

</li>

</ul>

<a class="close-link">

<i class="fa fa-times"></i>

</a>

</div>

</div>

<div class="inqbox-content">

<div class="tabs-container">

<ul class="nav nav-tabs tab-border-top-danger">

<li class="active"><a data-toggle="tab" href="#sectionA"><?= $this->lang->line("loans\_information"); ?></a></li>

<li><a data-toggle="tab" href="#sectionB"><?= $this->lang->line("loans\_misc\_fees"); ?></a></li>

<li><a data-toggle="tab" href="#sectionC"><?= $this->lang->line('loans\_attachments') ?></a></li>

<li><a data-toggle="tab" href="#sectionE"><?= $this->lang->line('guarantee') ?></a></li>

<li><a data-toggle="tab" href="#sectionF">Payment Schedule</a></li>

</ul>

<div class="tab-content">

<div id="sectionA" class="tab-pane fade in active">

<div style="text-align: center">

<h3><?= $this->lang->line("loans\_information"); ?></h3>

<div id="required\_fields\_message"><?php echo $this->lang->line('common\_fields\_required\_message'); ?></div>

<ul id="error\_message\_box"></ul>

</div>

<!--

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('loans\_type') . ':', 'loan\_type', array('class' => 'wide')); ?></label>

<div class="col-sm-10">

<?php echo form\_dropdown("loan\_type", $loan\_types, $loan\_info->loan\_type\_id, "id='loan\_type' class='form-control'"); ?>

<input type="hidden" id="loan\_type\_id" name="loan\_type\_id" value="<?= $loan\_info->loan\_type\_id; ?>" />

</div>

</div>

<div class="hr-line-dashed"></div>

-->

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('customers\_customer') . ':', 'inp-customer', array('class' => 'wide required')); ?></label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'inp-customer',

'id' => 'inp-customer',

'value' => $loan\_info->customer\_name,

'class' => 'form-control',

'placeholder' => $this->lang->line('common\_start\_typing'),

'style' => 'display:' . ($loan\_info->customer\_id <= 0 ? "" : "none")

)

);

?>

<span id="sp-customer" style="display: <?= ($loan\_info->customer\_id > 0 ? "" : "none") ?>">

<?= $loan\_info->customer\_name; ?>

<span><a href="javascript:void(0)" title="Remove Customer" class="btn-remove-row"><i class="fa fa-times"></i></a></span>

</span>

<input type="hidden" id="customer" name="customer" value="<?= $loan\_info->customer\_id; ?>" />

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('loans\_account') . ':', 'account', array('class' => 'wide required')); ?></label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'account',

'id' => 'account',

'value' => $loan\_info->account,

'class' => 'form-control'

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('loans\_description') . ':', 'description', array('class' => 'wide')); ?></label>

<div class="col-sm-10">

<?php

echo form\_textarea(

array(

'name' => 'description',

'id' => 'description',

'value' => $loan\_info->description,

'rows' => '5',

'cols' => '17',

'class' => 'form-control'

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('loans\_amount') . ':', 'amount', array('class' => 'wide required')); ?></label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'amount',

'id' => 'amount',

'value' => $loan\_info->loan\_amount,

'class' => 'form-control',

'type' => 'number',

'step' => 'any',

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group" id="data\_1"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('loans\_apply\_date') . ':', 'apply\_date', array('class' => 'wide required')); ?></label>

<div class="col-sm-10">

<div class="input-group date">

<span class="input-group-addon"><i class="fa fa-calendar"></i></span>

<?php

echo form\_input(

array(

'name' => 'apply\_date',

'id' => 'apply\_date',

'value' => (isset($loan\_info->loan\_applied\_date) && $loan\_info->loan\_applied\_date > 0) ? date("m/d/Y", $loan\_info->loan\_applied\_date) : date("m/d/Y"),

'class' => 'form-control',

'type' => 'datetime'

)

);

?>

</div>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('loans\_agent') . ':', 'agent', array('class' => 'wide')); ?></label>

<div class="col-sm-10">

<?php if ($user\_info->person\_id === '1'): ?>

<?php echo form\_dropdown("sel\_agent", $employees, ($loan\_info->loan\_agent\_id > 0 ? $loan\_info->loan\_agent\_id : $user\_info->person\_id), "id='sel\_agent' class='form-control'"); ?>

<?php else: ?>

<?= ucwords($user\_info->first\_name . " " . $user\_info->last\_name); ?>

<?php endif; ?>

<!--

<?php echo isset($loan\_info->agent\_name) ? ucwords($loan\_info->agent\_name) : ucwords($user\_info->first\_name . " " . $user\_info->last\_name); ?>

-->

<input type="hidden" id="agent" name="agent" value="<?= ($loan\_info->loan\_agent\_id > 0 ? $loan\_info->loan\_agent\_id : $user\_info->person\_id) ?>" />

<input type="hidden" id="approver" name="approver" value="<?= $loan\_info->loan\_approved\_by\_id; ?>" />

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('loans\_status') . ':', 'status', array('class' => 'wide')); ?></label>

<div class="col-sm-10">

<?= $loan\_status; ?>

<input type="hidden" id="status" name="status" value="<?= $loan\_info->loan\_status; ?>" />

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('loans\_remarks') . ':', 'remarks', array('class' => 'wide')); ?></label>

<div class="col-sm-10">

<?php

echo form\_textarea(

array(

'name' => 'remarks',

'id' => 'remarks',

'value' => $loan\_info->remarks,

'rows' => '5',

'cols' => '17',

'class' => 'form-control'

)

);

?>

</div>

</div>

</div>

<div id="sectionB" class="tab-pane fade">

<div style="text-align: center">

<h3><?= $this->lang->line("loans\_misc\_fees"); ?></h3>

</div>

<table class="table" id="tbl-income-sources">

<thead>

<tr>

<th style="text-align: center; width: 1%">

<input type="checkbox" class="select\_all\_" />

</th>

<th style="text-align: center; width: 80%"><?= $this->lang->line("loans\_fee"); ?></th>

<th style="text-align: center; width: 20%"><?= $this->lang->line("loans\_amount"); ?></th>

</tr>

</thead>

<tbody>

<?php foreach ($misc\_fees as $misc\_fee): ?>

<tr>

<td>

<input type="checkbox" class="select\_" />

</td>

<td>

<input type="text" class="form-control" name="fees[]" value="<?= $misc\_fee[0]; ?>" />

</td>

<td>

<input type="number" class="form-control" step="any" name="amounts[]" value="<?= $misc\_fee[1]; ?>" />

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<button class="btn btn-primary" type="button" id="btn-add-row"><?= $this->lang->line("common\_add\_row"); ?></button>

<button class="btn btn-danger" type="button" id="btn-del-row"><?= $this->lang->line("common\_delete\_row"); ?></button>

</div>

<div id="sectionC" class="tab-pane fade">

<h3><?= $this->lang->line('loans\_attachments') ?></h3>

<div id="required\_fields\_message"><?php echo $this->lang->line('loans\_attachments\_message'); ?></div>

<div>

<ul class="list-inline" id="filelist">

<?php foreach ($attachments as $attachment): ?>

<li>

<a href="uploads/loan-<?= $loan\_info->loan\_id; ?>/<?= $attachment['filename']; ?>" target="\_blank" title="<?= $attachment['filename']; ?>"><img src="<?= $attachment['icon']; ?>" /></a>

<span class="close remove-file" data-file-id="<?= $attachment['id']; ?>" title="Remove this file"><i class="fa fa-times-circle"></i></span>

</li>

<?php endforeach; ?>

</ul>

</div>

<div id="progress" class="overlay"></div>

<div class="progress progress-task" style="height: 4px; width: 15%; margin-bottom: 2px; display: none">

<div style="width: 0%;" aria-valuemax="100" aria-valuemin="0" aria-valuenow="50" role="progressbar" class="progress-bar progress-bar-info"></div>

</div>

<div id="container">

<a id="pickfiles" href="javascript:;" class="btn btn-default" data-loan-id="<?= $loan\_info->loan\_id; ?>"><?= $this->lang->line("common\_browse"); ?></a>

</div>

</div>

<div id="sectionE" class="tab-pane fade in">

<div style="text-align: center">

<h3><?= $this->lang->line("guarantee"); ?></h3>

<div id="required\_fields\_message"><?php echo $this->lang->line('common\_fields\_required\_message'); ?></div>

<ul id="error\_message\_box"></ul>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_name') . ':', 'guarantee\_name', array('class' => 'wide required')); ?>

</label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'guarantee\_name',

'id' => 'guarantee\_name',

'value' => $guarantee\_info->name,

'class' => 'form-control'

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_type') . ':', 'guarantee\_type', array('class' => 'wide required')); ?>

</label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'guarantee\_type',

'id' => 'guarantee\_type',

'value' => $guarantee\_info->type,

'class' => 'form-control'

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_brand') . ':', 'guarantee\_brand', array('class' => 'wide required')); ?>

</label>

<div class='col-sm-10'>

<?php

echo form\_input(

array(

'name' => 'guarantee\_brand',

'id' => 'guarantee\_brand',

'value' => $guarantee\_info->brand,

'class' => 'form-control'

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_make') . ':', 'guarantee\_make', array('class' => 'wide required')); ?>

</label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'guarantee\_make',

'id' => 'guarantee\_make',

'value' => $guarantee\_info->make,

'class' => 'form-control'

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_serial') . ':', 'guarantee\_serial', array('class' => 'wide required')); ?>

</label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'guarantee\_serial',

'id' => 'guarantee\_serial',

'value' => $guarantee\_info->serial,

'class' => 'form-control'

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group" id="data\_1">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_price') . ':', 'guarantee\_price', array('class' => 'wide required')); ?>

</label>

<div class="col-sm-10">

<div class="input-group">

<?php

echo form\_input(

array(

'name' => 'guarantee\_price',

'id' => 'guarantee\_price',

'value' => $guarantee\_info->price,

'class' => 'form-control'

)

);

?>

</div>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_proof') . ':', 'guarantee\_proof', array('class' => 'wide')); ?>

</label>

<div class="col-sm-10">

<div class='form\_field'>

<ul class="sel-proof">

<?php if (is\_array($proofs)): ?>

<?php foreach ($proofs as $proof): ?>

<li><input type="hidden" name="proofs[]" value="<?= $proof["id"]; ?>" /><a href="<?= site\_url(); ?>uploads/loan-<?= $loan\_info->loan\_id; ?>/<?= $proof["filename"]; ?>" target="\_blank"><?= $proof["descriptions"]; ?></a></li>

<?php endforeach; ?>

<?php endif; ?>

</ul>

<a href="loans/attachments/<?= $loan\_info->loan\_id; ?>/proof" data-toggle='modal' data-target='#attachment\_modal' >Select from attachment</a>

</div>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_images') . ':', 'guarantee\_images', array('class' => 'wide')); ?>

</label>

<div class="col-sm-10">

<ul class="sel-images">

<?php if (is\_array($pimages)): ?>

<?php foreach ($pimages as $pimage): ?>

<li><input type="hidden" name="images[]" value="<?= $pimage["id"]; ?>" /><a href="<?= site\_url(); ?>uploads/loan-<?= $loan\_info->loan\_id; ?>/<?= $pimage["filename"]; ?>" target="\_blank"><?= $pimage["descriptions"]; ?></a></li>

<?php endforeach; ?>

<?php endif; ?>

</ul>

<a href="loans/attachments/<?= $loan\_info->loan\_id; ?>/images" data-toggle='modal' data-target='#attachment\_modal'>Select from attachment</a>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

<?php echo form\_label($this->lang->line('guarantee\_observations') . ':', 'guarantee\_observations', array('class' => 'wide')); ?>

</label>

<div class="col-sm-10">

<?php

echo form\_textarea(

array(

'name' => 'guarantee\_observations',

'id' => 'guarantee\_observations',

'value' => $guarantee\_info->observations,

'class' => 'form-control'

)

);

?>

</div>

</div>

</div>

<div id="sectionF" class="tab-pane fade in">

<div style="text-align: center">

<h3>Payment Schedule</h3>

<div id="required\_fields\_message"><?php echo $this->lang->line('common\_fields\_required\_message'); ?></div>

<ul id="error\_message\_box"></ul>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">

Term:

</label>

<div class="col-sm-10">

<select type="text" name="term" class="form-control" style="display: inline; width: 30%" >

<?php

$opt = "";

foreach($paymenterm as $row){

$opt .= "<option value=".$row['paymenterm\_id'].">".$row['payment\_term']."</option>";

}

echo $opt;

?>

</select>

<select class="form-control" style="display: inline; width: 40%" name="term\_period">

<option value="day" <?= $c\_payment\_scheds["term\_period"] === "day" ? 'selected="selected"' : ''; ?>>Day</option>

<option value="week" <?= $c\_payment\_scheds["term\_period"] === "week" ? 'selected="selected"' : ''; ?>>Week</option>

<option value="month" <?= $c\_payment\_scheds["term\_period"] === "monthly" ? 'selected="selected"' : ''; ?>>Month</option>

<option value="year" <?= $c\_payment\_scheds["term\_period"] === "yearly" ? 'selected="selected"' : ''; ?>>Year</option>

</select>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

Schedule of Payment:

</label>

<div class="col-sm-10">

<select class="form-control" name="payment\_sched" style="display: inline; width: 30%">

<option value="daily" <?= $c\_payment\_scheds["payment\_sched"] === "daily" ? 'selected="selected"' : ''; ?>>Daily</option>

<option value="weekly" <?= $c\_payment\_scheds["payment\_sched"] === "weekly" ? 'selected="selected"' : ''; ?>>Weekly</option>

<option value="biweekly" <?= $c\_payment\_scheds["payment\_sched"] === "biweekly" ? 'selected="selected"' : ''; ?>>Bi-weekly</option>

<option value="monthly" <?= $c\_payment\_scheds["payment\_sched"] === "monthly" ? 'selected="selected"' : ''; ?>>Monthly</option>

<option value="bimonthly" <?= $c\_payment\_scheds["payment\_sched"] === "bimonthly" ? 'selected="selected"' : ''; ?>>Bi-Monthly</option>

<option value="yearly" <?= $c\_payment\_scheds["payment\_sched"] === "yearly" ? 'selected="selected"' : ''; ?>>Yearly</option>

</select>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

Interest Rate:

</label>

<div class='col-sm-10'>

<input type="text" class="form-control" name="interest\_rate" style="display: inline; width: 30%" value="<?= $c\_payment\_scheds["interest\_rate"] ?>" />%

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

Penalty:

</label>

<div class="col-sm-10">

<input type="text" class="form-control" name="penalty" style="display: inline; width: 30%" value="<?= $c\_payment\_scheds["penalty"] ?>" />%

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">

Schedules:

</label>

<div class="col-sm-10">

<table class="table table-bordered" id="tbl-payment-sched">

<thead>

<tr>

<th style="text-align: center; width: 1%">

<input type="checkbox" class="payment\_select\_all\_" />

</th>

<th style="text-align: center;">Schedule</th>

<th style="text-align: center;">Balance</th>

<th style="text-align: center;">Interest</th>

<th style="text-align: center;">Payment</th>

</tr>

</thead>

<tbody>

<?php if (count($c\_payment\_scheds["payment\_breakdown"]["balance"]) > 0): ?>

<?php for ($i = 0; $i < count($c\_payment\_scheds["payment\_breakdown"]["balance"]); $i++): ?>

<tr>

<td><input type='checkbox' class='payment\_select\_' /></td>

<td style="width:30%">

<div class="input-group date">

<span class="input-group-addon"><i class="fa fa-calendar"></i></span>

<input type="text" name="payment\_date[]" class="form-control" value="<?= $c\_payment\_scheds["payment\_breakdown"]["schedule"][$i]; ?>" />

</div>

</td>

<td><input type="number" name="payment\_balance[]" class="form-control" value="<?= $c\_payment\_scheds["payment\_breakdown"]["balance"][$i]; ?>"/></td>

<td><input type="number" name="payment\_interest[]" class="form-control" value="<?= $c\_payment\_scheds["payment\_breakdown"]["interest"][$i]; ?>"/></td>

<td><input type="number" name="payment\_amount[]" class="form-control" value="<?= $c\_payment\_scheds["payment\_breakdown"]["amount"][$i]; ?>"/></td>

</tr>

<?php endfor; ?>

<?php else: ?>

<tr>

<td><input type='checkbox' class='payment\_select\_' /></td>

<td style="width:30%">

<div class="input-group date">

<span class="input-group-addon"><i class="fa fa-calendar"></i></span>

<input type="text" name="payment\_date[]" class="form-control" value="" />

</div>

</td>

<td><input type="number" name="payment\_balance[]" class="form-control" value=""/></td>

<td><input type="number" name="payment\_interest[]" class="form-control" value=""/></td>

<td><input type="number" name="payment\_amount[]" class="form-control" value=""/></td>

</tr>

<?php endif; ?>

</tbody>

</table>

<button class="btn btn-primary" type="button" id="btn-add-row-payment"><?= $this->lang->line("common\_add\_row"); ?></button>

<button class="btn btn-danger" type="button" id="btn-del-row-payment"><?= $this->lang->line("common\_delete\_row"); ?></button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="row">

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

<div class="inqbox float-e-margins">

<div class="inqbox-title">

<h5>

<?php echo $this->lang->line("payments\_info"); ?>

</h5>

<div class="inqbox-tools">

<a class="collapse-link">

<i class="fa fa-chevron-up"></i>

</a>

<a class="close-link">

<i class="fa fa-times"></i>

</a>

</div>

</div>

<div class="inqbox-content">

<div style="text-align: center">

<div id="required\_fields\_message"><?php echo $this->lang->line('common\_fields\_required\_message'); ?></div>

<ul id="error\_message\_box"></ul>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">

Current Balance:

</label>

<div class="col-sm-10">

<span id="sp-current-balance"><?php echo $loan\_info->loan\_balance; ?></span>

<input type="hidden" id="current\_balance" name="current\_balance" value="<?= $loan\_info->loan\_balance; ?>" />

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group" id="data\_1"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('payments\_date') . ':', 'payment\_date', array('class' => 'wide required')); ?></label>

<div class="col-sm-10">

<div class="input-group date">

<span class="input-group-addon"><i class="fa fa-calendar"></i></span>

<?php

echo form\_input(

array(

'name' => 'date\_paid',

'id' => 'date\_paid',

'value' => date("m/d/Y"),

'class' => 'form-control',

)

);

?>

</div>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('payments\_amount') . ':', 'teller', array('class' => 'wide')); ?></label>

<div class="col-sm-10">

<?php

echo form\_input(

array(

'name' => 'paid\_amount',

'id' => 'paid\_amount',

'value' => '',

'class' => 'form-control',

'type' => 'number',

'step' => 'any',

)

);

?>

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('payments\_teller') . ':', 'teller', array('class' => 'wide')); ?></label>

<div class="col-sm-10">

<?php echo ucwords($user\_info->first\_name . " " . $user\_info->last\_name); ?>

<input type="hidden" id="teller" name="teller" value="<?= $user\_info->person\_id; ?>" />

</div>

</div>

<div class="hr-line-dashed"></div>

<div class="form-group"><label class="col-sm-2 control-label"><?php echo form\_label($this->lang->line('common\_remarks') . ':', 'remarks', array('class' => 'wide')); ?></label>

<div class="col-sm-10">

<?php

echo form\_textarea(

array(

'name' => 'remarks',

'id' => 'remarks',

'value' => '',

'rows' => '5',

'cols' => '17',

'class' => 'form-control'

)

);

?>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="form-group">

<div class="col-sm-offset-2">

<button type="button" class="btn btn-default" data-dismiss="modal" id="btn-close"><?= $this->lang->line("common\_close"); ?></button>

<button id="btn-approve" class="btn btn-success" type="button"><?= $this->lang->line('loans\_approve'); ?></button>

<?php if ($loan\_info->loan\_id > 0): ?>

<a href="<?= site\_url("loans/" . ($loan\_info->loan\_type\_id > 0 ? "generate\_breakdown" : "fix\_breakdown") . "/$loan\_info->loan\_id"); ?>" target="\_blank" id="btn-sched" class="btn btn-warning"><?= $this->lang->line('loans\_breakdown'); ?></a>

<a href="<?= site\_url("loans/print\_disclosure/$loan\_info->loan\_id"); ?>" target="\_blank" id="btn-break-gen" class="btn btn-primary" type="button"><?= $this->lang->line('loans\_disclosure'); ?></a>

<?php endif; ?>

<button id="btn-edit" class="btn btn-danger" type="button"><?= $this->lang->line('common\_edit'); ?></button>

<?php

echo form\_submit(

array(

'name' => 'submit',

'id' => 'btn-save',

'value' => $this->lang->line('common\_save'),

'class' => 'btn btn-primary'

)

);

?>

</div>

</div>

</div>

<!-- Modal -->

<div class="modal fade" id="attachment\_modal" role="dialog">

<div class="modal-dialog">

<div class="modal-content">

fdsfas

</div>

<!-- /.modal-content -->

</div>

<!-- /.modal-dialog -->

</div>

<!-- /.modal -->

<?php

echo form\_close();

?>

<?php $this->load->view("partial/footer"); ?>

<!-- Date picker -->

<script src="js/plugins/datapicker/bootstrap-datepicker.js"></script>

<script src="<?php echo base\_url(); ?>js/loan.js?v=<?= time(); ?>"></script>

<script type='text/javascript'>

//validation and submit handling

$(document).ready(function () {

$('.input-group.date').datepicker({

todayBtn: "linked",

keyboardNavigation: false,

forceParse: false,

calendarWeeks: true,

autoclose: true

});

$(document).on("change, keyup", "#amount", function () {

$("#sp-current-balance").html($(this).val());

$("#current\_balance").val($(this).val());

});

$(document).on("click", ".remove-file", function () {

var el = $(this);

$.ajax({

url: '<?= site\_url('loans/remove\_file'); ?>',

data: {file\_id: el.data("file-id"), softtoken: $("#token\_hash").val()},

type: 'post',

dataType: 'json',

success: function (data) {

el.parent().remove();

},

error: function () {

;

}

});

});

$("#btn-add-row").click(function () {

$(".select\_all\_").prop("checked", false);

var rowCount = $('#tbl-misc-fees tr').length;

if (rowCount > 1)

{

$("#tbl-income-sources tbody").append("<tr>" + $('#tbl-income-sources tr:last').html() + "</tr>");

}

else

{

$("#tbl-income-sources tbody").append("<tr><td><input type='checkbox' class='select\_' /></td><td><input type='text' class='form-control' name='sources[]' /></td><td><input type='number' class='form-control' name='values[]' /></td></tr>");

}

});

$("#btn-del-row").click(function () {

$('.select\_').each(function () {

if ($(this).is(":checked"))

{

$(this).parent().parent().remove();

}

});

});

$("#loan\_type").change(function () {

$("#loan\_type\_id").val($(this).val());

});

$("#sel\_agent").change(function () {

$("#agent").val($(this).val());

});

$("#btn-approve").click(function () {

$("#approver").val($("#user\_info").val());

$("#loan\_form").submit();

});

if ($("#agent").val() <= 0)

{

$("#agent").val($("#user\_info").val());

}

if ($("#loan\_id").val() > -1)

{

$("#loan\_form input, textarea").prop("readonly", true);

$("#loan\_form select").prop("disabled", true);

$("#btn-add-row").prop("disabled", true);

$("#btn-del-row").prop("disabled", true);

$("#btn-save").hide();

if ($("#status").val() !== "approved")

{

$("#btn-approve").show();

}

else

{

$("#btn-approve").hide();

}

$("#btn-break-gen").show();

$("#btn-edit").show();

$("#btn-edit").click(function () {

$("#btn-save").show();

$(this).hide();

$("#loan\_form input, textarea").prop("readonly", false);

$("#loan\_form select").prop("disabled", false);

$("#btn-add-row").prop("disabled", false);

$("#btn-del-row").prop("disabled", false);

$("#btn-save").show();

});

}

else

{

$("#btn-approve").hide();

$("#btn-break-gen").hide();

$("#btn-edit").hide();

}

$(document).on("click", ".btn-remove-row", function () {

$("#sp-customer").hide();

$("#sp-customer").html("");

$("#inp-customer").val("");

$("#inp-customer").show();

$("#customer").val("");

});

$('#inp-customer').autocomplete({

serviceUrl: '<?php echo site\_url("loans/customer\_search"); ?>',

onSelect: function (suggestion) {

$("#account").val(suggestion.data);

$("#customer").val(suggestion.data);

$("#sp-customer").html(suggestion.value + ' <span><a href="javascript:void(0)" title="Remove Customer" class="btn-remove-row"><i class="fa fa-times"></i></a></span>');

$("#sp-customer").show();

$("#inp-customer").hide();

}

});

var settings = {

ignore: "",

invalidHandler: function (form, validator) {

set\_feedback("Error: Please correct all the required fields", 'error\_message', true);

},

submitHandler: function (form) {

$(form).ajaxSubmit({

success: function (response) {

post\_loan\_form\_submit(response);

if ($("#loan\_id").val() > 0)

{

setTimeout(function () {

location.reload()

}, 5000);

}

},

dataType: 'json',

type: 'post'

});

},

rules: {

account: "required",

amount: "required",

"inp-customer": "required"

},

messages: {

account: "<?php echo $this->lang->line('loans\_account\_required'); ?>",

amount: "<?php echo $this->lang->line('loans\_amount\_required'); ?>",

"inp-customer": "<?php echo $this->lang->line('loans\_customer\_required'); ?>"

}

};

$('#loan\_form').validate(settings);

function post\_loan\_form\_submit(response) {

if (!response.success)

{

set\_feedback(response.message, 'error\_message', true);

}

else

{

set\_feedback(response.message, 'success\_message', false);

}

$('#loan\_form').attr("action", "<?= site\_url(); ?>loans/save/" + response.loan\_id);

}

$("#btn-add-row-payment").click(function () {

$(".payment\_select\_all\_").prop("checked", false);

var rowCount = $('#tbl-payment-sched tr').length;

if (rowCount > 1)

{

$("#tbl-payment-sched tbody").append("<tr>" + $('#tbl-payment-sched tr:last').html() + "</tr>");

}

else

{

$("#tbl-payment-sched tbody").append("<tr><td><input type='checkbox' class='payment\_select\_' /></td><td><input type='date' class='form-control' name='payment\_date[]' /></td><td><input type='number' class='form-control' name='payment\_balance[]' /></td><td><input type='number' class='form-control' name='payment\_interest[]' /></td><td><input type='number' class='form-control' name='payment\_amount[]' /></td></tr>");

}

$('.input-group.date').datepicker({

todayBtn: "linked",

keyboardNavigation: false,

forceParse: false,

calendarWeeks: true,

autoclose: true

});

});

$("#btn-del-row-payment").click(function () {

$('.payment\_select\_').each(function () {

if ($(this).is(":checked"))

{

$(this).parent().parent().remove();

}

});

});

});

</script>