***CAKEPHP NOTE***

How to install cake php?

Step 1:- app -> config -> eidt database file with only database.php

Step 2:- app-> config -> open core.php -> edit Security.salt some value is remove and add -> edit Security.cipherSeed some value is remove and add.

Echo $this->Form->create(‘Search’, array('enctype' => 'multipart/form-data')

) // create a form. ‘search’ will be model name.

Echo $this->Form->end(); //end form.

Echo $this->Form->end(‘Save’); // form end and submit save button create.

Echo $this->Form->submit(‘Save’);

Echo $this->Form->text(‘name’,array(‘class’=>’name\_class’)) // text box create.

Echo $this->Form->input(‘name’) // label and text box create.

Echo $this->Form->input(‘name’,array(‘label’=>’Your Name’)); // text box and label display.

Echo $this->Form->input(‘Model.name’);

Echo $this->Form->password(‘password’); // set password.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Select Box dropdown \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$arr= array(‘s’=>’small’,’m’=>’medium’,’l’=>’large’);

Echo $this->Form->input(‘size’,$arr); // dropdown menu, select box.

OR

Echo $this->Form->select(‘size’, $arr, 'empty' => '(choose one)'

);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Select group box \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$options = array(

'Group 1' => array(

'Value 1' => 'Label 1',

'Value 2' => 'Label 2'

),

'Group 2' => array(

'Value 3' => 'Label 3'

)

);

echo $this->Form->select('field', $options);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Echo $this->Form->checkbox(‘name’,array(‘value’=>’Y’)); // checkbox define.

Echo $this->form->hidden(‘id’,array(‘value’=>’2’)); // hidden value = 2 set.

Echo $this->Form->textarea(‘message’,array(‘rows’=>’5’,’cols’=>’5’));

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Radio Button \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$options=array(‘m’=>’male’,’f’=>’Female’);

$attributes=array(‘legend’=>false);

Echo $this->Form->radio(‘gender’,$options,$attributes);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Year Dorpdown\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

echo $this->Form->input('field', array(

'options' => array(1, 2, 3, 4, 5),

'empty' => '(choose one)'

));

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Select Group By\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$options = array(

'Group 1' => array(

'Value 1' => 'Label 1',

'Value 2' => 'Label 2'

),

'Group 2' => array(

'Value 3' => 'Label 3'

)

);

echo $this->Form->select('field', $options);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Echo $this->Form->file(‘photo’);

echo $this->Form->year('purchased', 2000, date('Y')); // year listing.

echo $this->Form->month('mob'); // month listing

echo $this->Form->day('created'); // day listing.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Image Include\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

echo $this->Html->image('postheadericon.png', array('alt' => 'CakePHP'));

OR

echo $this->Html->image("postheadericon.png", array(

"alt" => "Brownies",

'url' => array('controller' => 'students', 'action' => 'add')

)); //with anchor tag

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Link \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

echo $this->Html->link('Enter','add');// without pass a value

echo $this->Html->link('Enter1',array('controller'=>'students','action'=>'add',5));

// pass a value of a controller.

echo $this->Html->link('delete',array('controller'=>'students','action'=>'add'),null,'are u?');

// javascript message alert.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Nested List \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$list = array(

'Languages' => array(

'English' => array(

'American',

'Canadian',

'British',

),

'Spanish',

'German',

)

);

echo $this->Html->nestedList($list);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Table Header \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

echo $this->Html->tableHeaders(array('Date', 'Title', 'Active'));

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Table cells \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

echo $this->Html->tableCells(array(

array('Jul 7th, 2007', 'Best Brownies', 'Yes'),

array('Jun 21st, 2007', 'Smart Cookies', 'Yes'),

array('Aug 1st, 2006', 'Anti-Java Cake', 'No'),

));

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*How to select a drop down value in controller\*\*\*\*\*\*\*\*\*/

$arr\_di=$this->WbDistrict->find('list',array('fields'=>array('id','dist\_name')));

$this->set('arr\_di',$arr\_di);

Note- Where “WbDistrict” is a table name.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Image Upload a component \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

<?php

/\*\*

\* This is Image component to resize image

\*/

class ImageComponent extends Component {

var $cacheDir = 'imagecache'; // relative to $uploadDir path

var $uploadDir = 'uploads/';

/\*\*

\* Automatically upload an image and returns new image name

\*

\* @param array $imageArray Full image array.

\* @param string $imagePath Path to the image file.

\* @param string $rename Image new name, default is null

\* @return mixed Either string or false, depends on valid image.

\* @access public

\*/

public function upload($imageArray, $imagePath, $rename = null) {

//pr($imageArray);exit;

if (is\_uploaded\_file($imageArray['tmp\_name'])) {

$filename = explode('.', $imageArray['name']);

$ext = strtolower(end($filename));

$imageArray['name'] = time() . ((!is\_null($rename)) ? $rename : str\_replace(" ", "-", $imageArray['name']));

if ($ext === 'png' || $ext === 'jpeg' || $ext === 'jpg') {

if (!is\_dir($imagePath)) {

mkdir($imagePath, 0777);

}

move\_uploaded\_file($imageArray['tmp\_name'], $imagePath . '/' . $imageArray['name']);

} else {

$this->Session->setFlash(\_\_("Invalid image type. Upload only .png image"));

return false;

}

} else {

$this->Session->setFlash(\_\_("Empty image file"));

return false;

}

return $imageArray['name'];

}

/\*\*

\* Automatically resizes an image and returns formatted IMG tag

\*

\* @param string $path Path to the image file, relative to the webroot/img/ directory.

\* @param integer $width Image of returned image

\* @param integer $height Height of returned image

\* @param boolean $aspect Maintain aspect ratio (default: true)

\* @param array $htmlAttributes Array of HTML attributes.

\* @param boolean $return Wheter this method should return a value or output it. This overrides AUTO\_OUTPUT.

\* @return mixed Either string or echos the value, depends on AUTO\_OUTPUT and $return.

\* @access public

\*/

function resize($path, $width, $height, $aspect = true, $crop = false, $cropvars = array(), $autocrop = false, $htmlAttributes = array(), $return = false, $webrootPath, $onlyPath = false) {

$types = array(1 => "gif", "jpeg", "png", "swf", "psd", "wbmp"); // used to determine image type

$fullpath = ROOT . DS . APP\_DIR . DS . WEBROOT\_DIR . DS . $this->themeWeb . $this->uploadDir;

//$url = $fullpath.$path;

//$url = "img/" . $path;

$url = $path;

if (!($size = getimagesize($url)))

return; // image doesn't exist

if ($width == $size[0] && $height == $size[1]) {

$webrootPath = FULL\_BASE\_URL . '/' . $webrootPath;

$newPath = explode("$this->uploadDir", $path);

$relfile = $webrootPath . $this->uploadDir . $newPath[1];

} else {

if ($autocrop) {

$multiplier = 1.0;

while (($width \* $multiplier < $size[0]) && ($height \* $multiplier < $size[1])) {

$multiplier += .01;

}

// make SURE we don't run over

$multiplier -= .01;

$cropw = floor($width \* $multiplier);

$croph = floor($height \* $multiplier);

//echo("$cropw $croph $width $height $multiplier $size[0] $size[1] <br />");

///die();

$xindent = ($size[0] - $cropw) / 2.0;

$yindent = ($size[1] - $croph) / 2.0;

$startx = floor($xindent);

$endx = $size[0] - ceil($xindent);

$starty = floor($yindent);

$endy = $size[1] - ceil($yindent);

//echo("$xindent, $yindent -> leads to: $startx, $starty start and $endx, $endy end");

//die();

$cropvars = array($startx, $starty, $endx, $endy);

}

if (($width > $size[0] || $height > $size[1]) && $autocrop) {

$multiplier = 1.0;

while (($width \* $multiplier >= $size[0]) || ($height \* $multiplier >= $size[1])) {

$multiplier -= .01;

}

$cropw = floor($width \* $multiplier);

$croph = floor($height \* $multiplier);

//echo("$cropw $croph $width $height $multiplier $size[0] $size[1] <br />");

///die();

$xindent = ($size[0] - $cropw) / 2.0;

$yindent = ($size[1] - $croph) / 2.0;

$startx = floor($xindent);

$endx = $size[0] - ceil($xindent);

$starty = floor($yindent);

$endy = $size[1] - ceil($yindent);

//echo("$xindent, $yindent -> leads to: $startx, $starty start and $endx, $endy end");

//die();

$cropvars = array($startx, $starty, $endx, $endy);

}

// check that user supplied full start and stop coordinates

if (sizeof($cropvars) == 4) {

if ($cropvars[0] > $size[0] || $cropvars[1] > $size[1] || $cropvars[2] > $size[0] || $cropvars[3] > $size[1] || $cropvars[0] < 0 || $cropvars[1] < 0 || $cropvars[2] < 0 || $cropvars[3] < 0) {

$crop = false;

}

} else {

$crop = false;

}

// temporarily set size to this for aspect checking

if ($crop) {

$tempsize = array($size[0], $size[1]);

$size[0] = $cropvars[2] - $cropvars[0];

$size[1] = $cropvars[3] - $cropvars[1];

}

if ($aspect) { // adjust to aspect

if (($size[1] / $height) > ($size[0] / $width)) // $size[0]:width, [1]:height, [2]:type

$width = ceil(($size[0] / $size[1]) \* $height);

else

$height = ceil($width / ($size[0] / $size[1]));

}

// set size back

if ($crop) {

$size[0] = $tempsize[0];

$size[1] = $tempsize[1];

}

if ($crop) {

$cropstring = $cropvars[0] . $cropvars[1] . $cropvars[2] . $cropvars[3] . '\_';

} else {

$cropstring = '';

}

$webrootPath = FULL\_BASE\_URL . '/' . $webrootPath;

$relfile = $webrootPath . $this->themeWeb . $this->uploadDir . $this->cacheDir . '/' . $width . 'x' . $height . '\_' . $cropstring . basename($path); // relative file

$cachefile = $fullpath . $this->cacheDir . DS . $width . 'x' . $height . '\_' . $cropstring . basename($path); // location on server

if (file\_exists($cachefile)) {

$csize = getimagesize($cachefile);

$cached = ($csize[0] == $width && $csize[1] == $height); // image is cached

if (@filemtime($cachefile) < @filemtime($url)) // check if up to date

$cached = false;

} else {

$cached = false;

}

if (!$cached) {

$resize = ($size[0] > $width || $size[1] > $height) || ($size[0] < $width || $size[1] < $height);

} else {

$resize = false;

}

if ($resize) {

$image = call\_user\_func('imagecreatefrom' . $types[$size[2]], $url);

if ($crop) {

if (function\_exists("imagecreatetruecolor") && ($tempcrop = imagecreatetruecolor($cropvars[2] - $cropvars[0], $cropvars[3] - $cropvars[1]))) {

imagecopyresampled($tempcrop, $image, 0, 0, $cropvars[0], $cropvars[1], $cropvars[2] - $cropvars[0], $cropvars[3] - $cropvars[1], $cropvars[2] - $cropvars[0], $cropvars[3] - $cropvars[1]);

} else {

$temp = imagecreate($cropvars[2] - $cropvars[0], $cropvars[3] - $cropvars[1]);

imagecopyresized($tempcrop, $image, 0, 0, $cropvars[0], $cropvars[1], $cropvars[2] - $cropvars[0], $cropvars[3] - $cropvars[1], $size[0], $size[1]);

}

$image = $tempcrop;

}

$size[0] = $cropvars[2] - $cropvars[0];

$size[1] = $cropvars[3] - $cropvars[1];

if (function\_exists("imagecreatetruecolor") && ($temp = imagecreatetruecolor($width, $height))) {

imagecopyresampled($temp, $image, 0, 0, 0, 0, $width, $height, $size[0], $size[1]);

} else {

$temp = imagecreate($width, $height);

imagecopyresized($temp, $image, 0, 0, 0, 0, $width, $height, $size[0], $size[1]);

}

if ($types[$size[2]] == "jpeg") {

imagejpeg($temp, $cachefile, 90);

} else {

call\_user\_func("image" . $types[$size[2]], $temp, $cachefile);

}

imagedestroy($image);

imagedestroy($temp);

}

//return $this->output(sprintf($this->Html->image, $relfile, $this->Html->\_parseAttributes($htmlAttributes, null, '', ' ')), $return);

}

if ($onlyPath) {

return $relfile;

}

$htmlAttributes['src'] = $relfile;

return $this->imageAttributes($htmlAttributes);

//return '<img src="' . $relfile . '" alt="" />';

}

private function imageAttributes($htmlAttributes) {

$imgAtr = array();

if (!isset($htmlAttributes['src']) && empty($htmlAttributes['src']))

return false;

foreach ($htmlAttributes as $attribute => $value) {

$imgAtr[] = "$attribute=" . '"' . $value . '"';

}

return "<img " . implode(' ', $imgAtr) . " />";

}

}

?>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* URL \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

WWW\_ROOT // root path. Exp = [\\lserver\Cramdude\web\app\webroot](file:///\\lserver\Cramdude\web\app\webroot)

/\*\*\*\*\*\*\*\*\*\*\*\*\*Form create different controller \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

echo $this->Form->create('DepartmentUniversity', array(

'url' => array('controller' => 'departments', 'action' => 'edit\_university\_department', 'admin' => true)

));

//end

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Last Insertid \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$university\_id = $this->University->getLastInsertId();

//end

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TinyMCE Editor \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

tinymce.init({

selector: "textarea",

plugins: [

"advlist autolink link image lists charmap print preview hr anchor pagebreak spellchecker",

"searchreplace wordcount visualblocks visualchars code fullscreen insertdatetime nonbreaking",

"save table contextmenu directionality emoticons template paste textcolor"

],

toolbar: "undo redo | styleselect | bold italic underline | alignleft aligncenter alignright alignjustify | bullist numlist outdent indent | forecolor backcolor | link image fullscreen preview code",

convert\_urls: false

});

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Create a own library \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

App/lib/CommonLibrary.php

<?php

class CommonLibrary {

function sayHello(){

return "hahaha";

}

}

?>

App/config/bootstrap.php

$test = App::uses('CommonLibrary', 'Lib');

OR

App/controller/controllername.php

uses() is a statement you place at the beginning of the file

$test = App::uses('CommonLibrary', 'Lib');

class ContentsController extends AppController {}

call library for any section –

$test = new CommonLibrary();

echo $test->sayHello();