**Installation instructions:**

Copy the files your server. Just include the framework.php file in your codes, that’s it!

Just configure the framework/config.php as per your needs.

You can add your own php functions in the file framework/function.php

The framework/classes folder is a collection of classes and related functions. You can add functions or classes as per your need and add the objects of the class in config.php and a requirement of the class in framework.php

In the entire framework security has been taken special care of.

**Validation:**

These are the functions of the validation class currently having an object as $val. But this can be easily customized as per your requirement from the config.php file in framework directory.

1. email

Arguments: 1

Returns true if string is email or else false.

Usage: $val->email(“some\_email@somesite.com”);

2. username

Arguments: 1

Returns true if string is valid username or else false.

Usage: $val-> username(“username\_name”);

3. url

Arguments: 1

Returns true if string is url or else false.

Usage: $val->url (“http://website.com”);

4. creditcard

Arguments: 1

Returns true if string is credit card number or else false.

Usage: $val->creditcard (“1738292928284637”);

5. date

Arguments: 2

First argument is the string to be validated as date and second is the type of date.

Returns true if string is date or else false.

Usage: $val->date (“1989.05.12”,” YYYY.MM.DD”);

6. ip

Arguments: 1

Returns true if string is image or else false.

Usage: $val->ip(“192.168.0.1”);

7. image

Arguments: 1

Returns true if string is image or else false.

Usage: $val->image(“uploads/new\_image.jpg”);

**PHP main functions:**

These functions are independent of any class. They are the basic php functions which are modified for security reasons. They are cleaned from xss or cross site scripting. They can be called directly.

1. get

Arguments: 1

Get data from url . Works same as $\_GET[‘parameter’] but secured and checks if get parameter exists.

Usage: $id=get(‘id’);

2. post

Arguments: 1

Get data from post . Works same as $\_POST[‘parameter’] but secured and checks if post parameter exists.

Usage: $email=post(‘email’);

3.cookie

Arguments: 1

Get data from a cookie. Works same as $\_COOKIE[‘parameter’] but secured and checks if cookie parameter exists.

Usage: $email=cookie(‘email’);

4. show

Arguments: 1

Works same as echo or print but secured and avoids printing harmful characters that can be a threat. Also can determine if the string has been posted since it returns a true value if displaying the string has been done successfully.

Usage: show(‘Any string or variable’);

**Security :**

Security is an essential part of PHP development. With the help of this class most php threats can be overcome. The object of security class in $sec but this can be changed from the config.php file in framework directory.

1. xss

Arguments: 1

Sanitizes string with this function can easily prevent all types of xss attempts. This can be used on any type of unknown or user inputted data.

$safe\_string=$sec->xss(‘Potentially harmful string’);

2. sqlinjection

Arguments: 1

Sanitizes string for prevention of sql injection that can harm the mysql database. Use of this functions can prevent any kind of harm to the mysql database . Must be used with inputs that are directly associated with the database.

Usage: $safe\_string =$sec-> sqlinjection(‘Potentially harmful string’);

3. ip

Arguments: 0

Checks the IP address of the client and detects if the client is a blacklisted IP address. This can be a malicious hacker or a bot trying to damage your server.

Returns true if the IP address is blacklisted and false if not.

Usage: $sec->ip();

4. crawler

Arguments: 0

Detects crawlers of search engines that may be crawling your website. It returns true if crawler is detected or else false. Using this crawlers can be easily prevented and stopped from accessing your website.

5. randchars

Arguments:1

Generates a random string of characters that are alphanumeric. It generates string of the length specified as an argument. The strings generated from this can be used as tokens or randomly generated passwords.

Usage:$num=$sec->randchars(‘6’); // outputs s4dIp9

**getinfo :**

This can be used as a tool to gather information from the client .

Object of this class is $gi and can be changed from config.php file in framework directiory

1. ip

Arguments:0

Returns the true IP address of the client computer. Detects proxy and gets the real IP address.

Usage:$ipaddress=$gi->ip();

2. browser

Arguments: 0

Get details of clients browser such as user agent , os, browser name and version

Returns an array of information

Usage:$browser=$gi->browser();

$browser[‘useragent’] will contain user agent

$browser[‘name’] will contain name

$browser[‘version’] will contain version

$browser[‘os’] will contain the operating system .

3. extension

Arguments: 1

Get extension of a particular file.

Usage: $gi->extension(‘somefile.dll’); // outputs “dll”

**Image**

This class contains various methods in which you can alter or change an image. It contains some basic image operations. Object of this class is $img but can be changed from the config.php file.

1. crop

Arguments: 3

Crops image to desired parameters.

First parameter is source image, second parameter is the target image which is outputted. The third parameter is an array consisting of how to crop the image. Pixel is the unit.

Usage:

$crop[‘width’]=100;

$crop[‘height’]=56;

$crop[‘top’]=10;

$crop[‘left’]=16;

$img->crop(‘source.jpg’,’target.jpg’,$crop);

2. orientation

Arguments: 1

Adjusts orientation of an image.

Usage: $img->orientation(‘my\_image.png’);

3.thumbnail

Arguments: 1

Creates a thumbnail of the image.

Usage: $img->thumbnail(‘image.jpg’);

4.qrcode

Arguments: 3

String of the data to be converted to qrcode.

Returns the source of the image . Default size is 300x300 but can be customized.

Second and third parameter is height and width of the qrcode image.

4. watermark

Arguments: 4

Puts a text watermark to an image .

First parameter is the source image, second is the text to be watermarked on the source image, third parameter is color of text and the fourth parameter is the font size of the watermarking text.