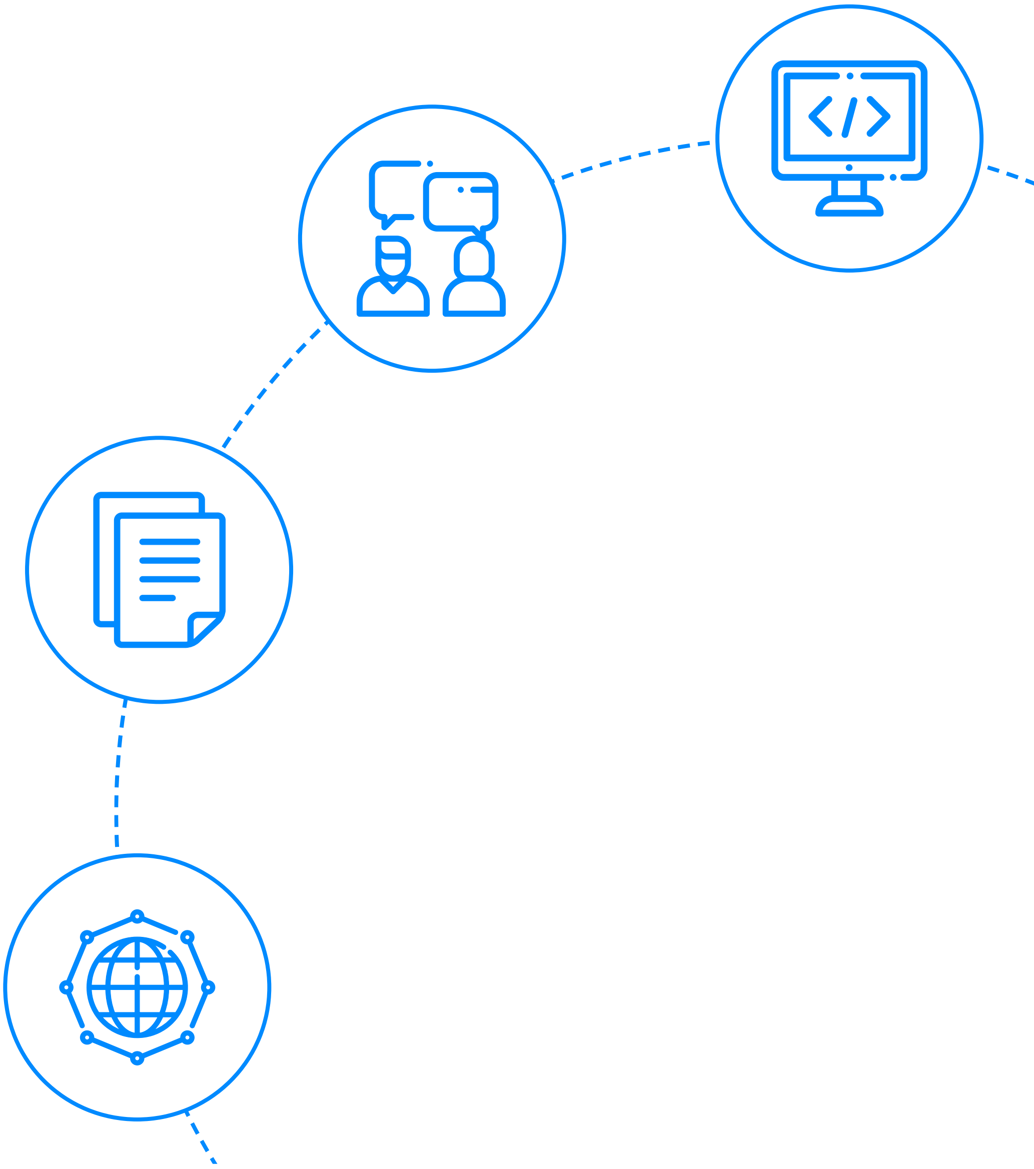
InterviewBit

CodeIgniter Interview Questions



To view the live version of the page, click here.

Contents

CodeIgniter Interview Questions for Freshers

CodeIgniter Interview Questions for Experienced

CodeIgniter Interview Questions for Experienced

(.....Continued)

19. What is the difference between Laravel and CodeIgniter?
20. List various databases supported by the CodeIgniter framework.
21. What is the work of anchor tag in CodeIgniter?
22. Explain CodeIgniter E-mail library. How to send an E-mail using CodeIgniter?
23. How to deal with Error handling in CodeIgniter?
24. Explain the default URL pattern used in CodeIgniter.
25. How you can add or load a model in CodeIgniter?
26. Explain the CodeIgniter framework.
27. How to pass an array from the controller to view in CodeIgniter?
28. Explain how to prevent CodeIgniter from CSRF(Cross Site Request Forgery).
29. What are the sessions in CodeIgniter? How to handle sessions in CodeIgniter?
30. Explain CodeIgniter folder structure.
31. What is the security parameter for XSS in CodeIgniter?
32. What is the default controller in CodeIgniter?
33. List all the auto-loadable resources available in CodeIgniter.
34. What do you mean by the controller in CodeIgniter?
35. Why is there a need to configure the URL routes?

Let's get Started

CodeIgniter is an open-source and powerful MVC(Model-View-Controller) based framework used for developing web applications on PHP. CodeIgniter provides libraries for connection with the database and to perform operations such as sending emails, uploading files, managing sessions, etc. It helps for PHP code simplification and brings out a fully interactive, dynamic website in a shorter span of time. The latest version of CodeIgniter is `CodeIgniter4 version v4.1.3` which was released on June 6, 2021.

Features of CodeIgniter:

CodeIgniter features will include the following:

- **Framework with a small footprint:**
  Source code for the CodeIgniter framework is nearly 2MB in size. It makes it easier to master CodeIgniter and how it works. In addition, it simplifies deploying and updating it.
- **Loosely coupled:**
  The built-in features are designed to work independently without depending too much on any other components. This makes it easier for maintaining and upgrading.
- **MVC Architecture:**
  CodeIgniter framework uses the MVC architectural design. MVC architecture separates the data, business logic, and presentation.
- **Blazing fast:**
  Users opt for applications that load very fast. If you have worked with few modern frameworks, you will realize that it takes less than one second to load immediately after installation. You can load CodeIgniter on average around less than 50ms.
- **Excellent and clear documentation:**
  CodeIgniter framework is having well-maintained documentation. Also, it has good tutorials, books, and answered forum questions on this. This implies whatever challenge you are facing, chances are someone has previously encountered the problem, solved it and the solution is available for you.
- **Extendable:**
  CodeIgniter includes several libraries and helpers out of the box. If what you want is not there or you wish to implement an existing feature your way, then you can do so easily by creating your own libraries, packages, helpers, etc. You are also permitted to create REST API in CodeIgniter.
- **Application-specific built-in components:**
  It has components for sending email, session management, database management, and much more.
- **Short learning curve:**
  CodeIgniter is easy to master for anyone who knows PHP. Within a shorter period, the student can learn CodeIgniter and start professional applications development using CodeIgniter.

Scope of CodeIgniter:

CodeIgniter is a PHP framework with a significantly small footprint, specifically built for developers who want to make use of a simple and graceful toolkit for creating completely featured and functional web applications. This application focus on enabling the users to develop the projects much faster and quicker than if you would have gone for writing your code from scratch. It is used for doing this by providing a set of libraries for the commonly needed applications and tasks.
It provides services for all the application modules to access the application database or external information resources in an OOP manner. Here, the model classes will have the functions that help us to insert, update, and retrieve information in the database.

CodeIgniter Interview Questions for Freshers

1. What is CodeIgniter?

CodeIgniter is an open-source and MVC-based framework used for web application development on PHP. This framework contains libraries, an easier interface with a logical structure to access these libraries, helpers, plug-ins, and other resources as well. It is easy to use compared to other PHP frameworks.

2. What are hooks in CodeIgnitor?

- CodeIgniter's hooks will provide a way to change the internal workings or framework functionalities without any need for hacking the core files. It permits script execution with a particular path within the CodeIgniter.
- We can globally enable/disable the hooks feature by setting the below-given item in the `application/config/config.php` file: `$config['enable_hooks'] = TRUE;`
- It is defined in `application/config/hooks.php` file. For example:

```
$hook['pre_controller'] = array(
'class' => 'MyHookClass',
'function' => 'Myhookfunction',
'filename' => 'MyHookClass.php',
'filepath' => 'hooks',
'params' => array('test', 'test1', 'webs')
);
```

In the above code example, the 'pre_controller' hook is called hook point. Various types of hook points are available in CodeIgniter.

3. What is an inhibitor in CodeIgniter?

An inhibitor in CodeIgniter is an error handler class. It will make use of PHP's native functions like `set_error_handler` , `set_exception_handler` , `register_shutdown_function` to handle parse errors, exceptions, and fatal errors.

4. How to check the CodeIgniter version?

There are 2 ways to check the CodeIgniter version.

- The first method is to run the following code:

```
<?php
   echo CI_VERSION;
?>
```

You can echo the constant value of CI_VERSION in the CodeIgniter controller or view file.

- The second method is to navigate to the `system/core/CodeIgniter.php` directory which stores the current version number of CodeIgniter in a global constant named 'CI_VERSION'. Open the file and have a look at the lines:

```
/**
 * CodeIgniter Version
 *
 * @var    string
 *
 */
  define('CI_VERSION', '4.1.3');
```

5. Explain the difference between helper and library in CodeIgniter.

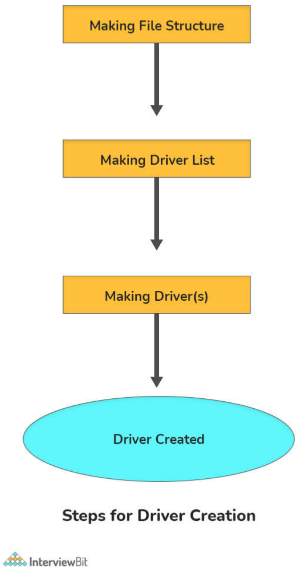
Helper	Library
Helper is a collection of common functions which we can use within Models, Views as well as in Controllers. Once we include the helper file, we can get access to the functions.	Library is a class that has a set of functions that permits for creating an instance of that class by `$this->load->library()` function.
It is not written in object-oriented format.	It is written in an object-oriented format.
It can be called in the same manner you call PHP functions.	You must create an object of the class to call library functions by using the `$this->library_name->method()` .
All built-in helper file names are suffixed with a word `_helper` (ex: `email_helper.php`).	All built-in library files do not have a specific suffix.

6. What is routing in CodeIgniter?

- Routing is a technique used in CodeIgniter, by which you can define your URLs based on the requirement instead of using the predefined URLs. So, whenever there is a request made and matches the URL pattern defined by us, it will automatically direct to the specified controller and function.
- A URL string and its corresponding controller class or method are in a one-to-one relationship here. The URI segments usually follow this pattern: `example.com/class/function/id/` . All routing rules are defined in the `application/config/routes.php` file of CodeIgniter.

7. What are drivers in CodeIgniter?

- A driver is a type of library that has a parent class and multiple child classes. These child classes can access their parent class, but they can't access their siblings.
- Drivers can be found in the system/libraries folder.
- There are three steps for creating a driver:
  - Making file structure
  - Making driver list
  - Making driver(s)



Steps for Driver Creation

8. How to link images from a view in CodeIgniter?

In Codeigniter, you can link `images/CSS/JavaScript` from a view by using the absolute path to the resources required with respect to the root folder as given below:

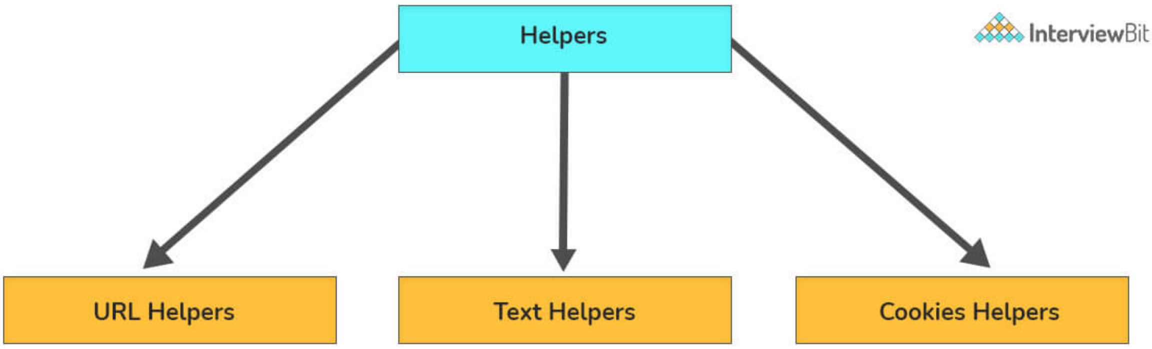
```
/css/styles.css
/js/query.php
/img/news/566.gpg
```

9. Why CodeIgniter is called a loosely based MVC framework?

Codeigniter is called a loosely based MVC framework because it does not need to obey a strict MVC pattern during application creation. It is not important to create a model, we can use only view and controllers for creating an application. In addition, one can modify CodeIgniter to utilize HMVC(Hierarchical Model View Controller) as well.

10. What is a helper in CodeIgniter?

- Helpers are the group of functions that are useful in assisting the user to perform specific tasks.
- There are three types of helper files. They are:
  - **URL helpers**: Used for creating the links.
  - **Text helpers**: Used for the formatting of text.
  - **Cookies helpers**: Used to read and manage cookies.

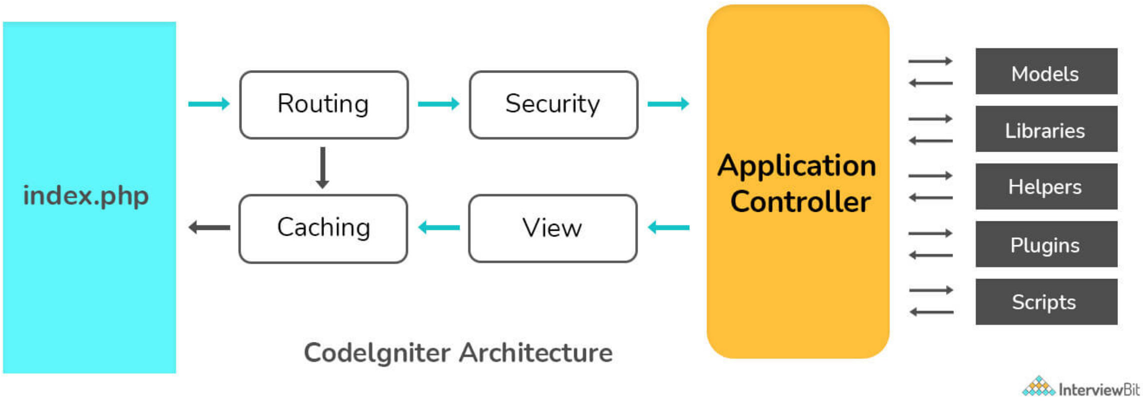


Types of Helpers in CodeIgniter

CodeIgniter Interview Questions for Experienced

11. Explain CodeIgniter Architecture.

- CodeIgniter is mainly designed to deliver high performance in less time within a good environment. For achieving this, each developing process is designed in a simplified manner.
- From the technical point of view, it is dynamically instantiated (libraries are loaded only on request which makes it light-weighted), has loose coupling (components depend very less on each other), and component singularity (each class and its functions are focused only on their purpose).
- **Data flow in CodeIgniter**: Below image represents that whenever a request is raised from the CodeIgniter application, firstly, it will go to the index.php file.
    - index.php is the default file of CodeIgniter. This file initializes the base resources.
    - The router determines what should be done with the information.
    - If the requested cache file exists, then the information is moved directly to the browser and ignores the further processes.
    - If the page requested by the user does not exist in the caching file, the HTTP request and data submitted will be passed under security check.
    - The application controller will load the models, libraries, helpers, plugins, and scripts required according to the request.
    - A view is used for fetching the data from the application controller that will be represented to the user, and they pass the data to the caching file to the fastest access for future requests.



CodeIgniter Architecture

12. How to load a helper in CodeIgniter?

- You need to load the helper files for using it. Once loaded, it will be globally available to your controller and views. They can be obtained at two places in CodeIgniter. A helper file will be searched by CodeIgniter in the `application/helpers` folder and if it is not available in that folder then it will check in the `system/helpers` folder.
- Helper file can be loaded by adding the following code to the constructor of the controller or inside any function that wants to use: `$this->load->helper('file_name');`
  Write your file name at the place of file_name.
- To load URL helper we can use the code given below: `$this->load->helper('url');`
- You are allowed to auto-load a helper if your application needs that helper globally by including it in the `application/config/autoload.php` file.
- Loading multiple helpers is also possible. For doing this, specify them in an array as given below:

```
$this->load->helper(
    array('helper1', 'helper2', 'helper3')
);
```

13. What are the advantages of CodeIgniter?

Few advantages of using CodeIgniter is given below:

- **Built-in libraries**: It comes with various types of default helpers for multiple things including strings, arrays, cookies, directories, file handling, and forms among others.
- **Data abstraction**: You can make use of the CodeIgniter database abstraction layer for creating, adding, deleting, and replacing statements in a hassle-free manner. This framework allows you to manage multiple connections using a single application.
- **Active Developer Community**: Bigger the community, the better the help you get. Newly graduated developers look forward to the framework's forum to get their doubts solved and learn about new things in the process. With so many people actively participating in it from around the world, your doubts will be solved within few hours. And due to the same reason, CodeIgniter documentation is 10 times bigger than any other framework.
- **Collaboration with Expression Engine**: The collaboration permits developers using CodeIgniter to use libraries and everything else provided by Expression Engine and vice versa. Because of this, developers will get few benefits like better parser class, improved built-in user authentication, and easy access to modular applications.
- **Security**: The security strength modification can be done according to your client's needs. These changes are made when the system is initialized by switching off the `magic_quotes_runtime` directive irrespective of the `register_globals` directive. You don't need to remove the slashes during information retrieval from the database. You can enable encryption of cookies, where you can handle databases and escape SQL queries directly.
- **Immigration Features**: Database schema update management is easier over different fields by using the migration aspect. It is an easier process to immigrate from the server to the server in Codeigniter.
- **Easy to Use**: It is easier to use compared to other popular frameworks such as Symfony, Zend framework, and Cake PHP.

14. Give the list of hooks available in CodeIgniter.

The list of available hook points are given below:

- `pre_system` : It is called initially during system execution.
- `pre_controller` : It is called immediately before any of the controllers being called. Example:

```
$hook['pre_controller'] = array(
   'class'    => 'ExampleClass',
   'function' => 'Examplefunction',
   'filename' => 'ExampleClass.php',
   'filepath' => 'hooks',
   'params'   => array('mango', 'apple', 'orange')
   );
```

- `post_controller_constructor` : It is called soon after instantiating your controller, but before any occurrence of the method call.
- `post_controller` : It is called immediately after the complete execution of the controller.
- `display_override` : It overrides the `_display()` method.
- `cache_override` : It enables calling of the user-defined method instead of `_display_cache()` method which is available in the Output Library. This permits you for using your own cache display mechanism.
- `post_system` : It is called soon after the final rendered page has been submitted to the web browser, at the end of system execution when the final data has been sent to the browser.

15. What is Command-Line Interface(CLI)? Why we use CLI in Codeigniter?

Command-Line Interface or CLI is a text-based interface for interacting with computers through a set of commands. We can use CLI in CodeIgniter for:

- Running your cron-jobs without `wget` or `curl` usage
- Make your cron-jobs inaccessible from being loaded in the URL(Uniform Resource Locator) by checking the value returned by `is_cli()`
- Make interactive "tasks" that can do various things such as set permissions, run backups, prune cache folders, etc.
- It helps to integrate CodeIgniter with applications in other languages. For example, a random C++ script can call a command and run code in your models.

16. What is meant by a library? How can you load a library in CodeIgniter?

- Libraries are packages created in PHP that give higher-level abstractions and thus contribute to faster development. This removes the necessity of focusing on small, minute details by taking care of those by themselves.
- Three methods are available to create a library:
  - Create an entirely new library
  - Extend native libraries
  - Replace native libraries
- To load a library in CodeIgniter, you have to include the below code inside a controller: `$this->load->library('class_name');`
- All pre-defined libraries developed by CodeIgniter can be obtained at the `system/libraries` directory.
- For loading multiple libraries at the same time, you can make use of the same code. But replace the parameter with an array for loading multiple libraries.

```
$this->load->library(array('library1', 'library2'));
```

17. What is CSRF token in CodeIgniter? How to set CSRF token?

- CSRF(Cross-Site Request Forgery) token is a randomly generated value that gets modified with every HTTP request sent by webform.
- A CSRF attack forces a browser of the logged-on victim for sending a forged HTTP request, including the session cookie of the victim and other information related to authorization, to a web application. A CSRF token is used for setting or activating the protection in CodeIgniter.
- CSRF token is saved in the user's session when it is added in the website form. When we submit the form, the website compares both submitted tokens and saved tokens in the session. If they are the same, a request is considered valid. When the page gets loaded token value will also be changed each time. Thus it becomes difficult for the hackers to identify the current token.
- To set CSRF, you have to set the corresponding config value as true in your `application/config/config.php` file.
  Syntax : `$config['csrf_protection'] = TRUE;`
  If you use the form helper, the form_open() method will automatically insert a hidden CSRF field in your forms.

18. How to extend the class in CodeIgniter?

You have to create a file with the name Example.php under `application/core/` directory and declare your class with the below code:

```
Class Example extends CI_Input {
     // Write your code here
}
```

19. What is the difference between Laravel and CodeIgniter?

Based on	Laravel	CodeIgniter
Database model	It is object-oriented.	It is relational object-oriented.
Built-in module	It comes along with a built-in module.	It does not come with a built-in module.
Structure	Follows MVC structure of filing with a command-line tool known as Artisan.	Follows the MVC structure but it provides easier boarding based on object-oriented programming.
Development and Template	It is a good option for front-end developers and it comes along with the Blade template engine.	It is easier to use and there is no template engine provided.
Utilized by	OctoberCMS, Laracasts	PyroCMS, Expression engine
Libraries	Provide their own official documentation which is very helpful.	Provides a lot of built-in functionality
Routing	Supports Explicit routing	Supports both Explicit and Implicit routing

20. List various databases supported by the CodeIgniter framework.

Following Databases are supported by the CodeIgniter framework:

- MySQL (version 5.1+) database that uses MySQL (deprecated), mysqli, and PDO drivers
- Oracle database that uses oci8 and PDO drivers
- PostgreSQL database that uses Postgre and PDO drivers
- ODBC database that uses ODBC and PDO drivers
- SQLite database that uses SQLite version 2, SQLite3 version 3, along with PDO drivers
- MS SQL database that uses Sqlsrv (version 2005 and above), MsSQL, and PDO drivers
- Interbase/Firebird database that uses iBase and PDO drivers
- CUBRID database that uses Cubridand PDO drivers

21. What is the work of anchor tag in CodeIgniter?

- Anchor tag creates a standard HTML anchor link based on the URL of your local site.
- **Syntax:**

```
anchor($uri = '', $title = '', $attributes = '')
```

- Here, $uri represents a URI string, $title represents an anchor title and $attributes represents an HTML attributes. It returns an HTML hyperlink (anchor tag) of string type.

The first parameter can have any segments you would like to append to the URL. These segments can be a string or an array.

The second parameter is the text that will be displayed with a link. The URL will be used in case you leave it blank.

The third parameter can contain an attribute list you would like added to the link. The attributes can be a string or an associative array.

- **Example:**

```
echo anchor('details/local/123', 'My Details', 'title="Details title"');
// Prints: <a href="http://example.com/index.php/details/local/123" title="Details titl
```

22. Explain CodeIgniter E-mail library. How to send an E-mail using CodeIgniter?

- Features of Email Class in CodeIgniter are given below:
    - Multiple protocols such as Mail, Sendmail, and SMTP
    - TLS and SSL Encryption for SMTP
    - CC and BCCs
    - Multiple recipients
    - Attachments
    - HTML or Plain-text email
    - Priorities
    - Word wrapping
    - BCC Batch Mode, enabling larger e-mail lists to be broken into smaller BCC batches
    - Email Debugging tools
- **Sending Email:**
  Sending an email is a simple process here. You can configure an email on the fly or set your preferences in the `app/Config/Email.php` file. A basic example for demonstrating how you might send email is given below:

```
$email = \Config\Services::email();
$email->setFrom('your@interviewbit.com', 'Your Name');
$email->setTo('someone@interviewbit.com');
$email->setCC('another@another-example.com');
$email->setBCC('them@their-example.com');
$email->setSubject('Email Test');
$email->setMessage('Testing the email class.');
$email->send();
```

23. How to deal with Error handling in CodeIgniter?

CodeIgniter enables you to develop error reporting into your applications by using the below-given functions. Also, it has a class dedicated to error logging that permits messages related to error and debugging to be saved as text files.

Functions related to error handling are:

- This function will display the error message provided by the `application/errors/errorgeneral.php` template.

```
show_error('message' [, int $statuscode= 500 ] )
```

- This function shows the 404 error message supplied to it by using the `application/errors/error404.php` template.

```
show_404('page' [, 'logerror'])
```

- This function permits you to write messages onto your log files. You must provide anyone among three "levels" in the first parameter that indicates the message type (debug, error, info), with the message itself in the second parameter.

```
log_message('level', 'message')
```

24. Explain the default URL pattern used in CodeIgniter.

- CodeIgniter will make use of a "segment-based" approach instead of a "query string-based" approach.
- CodeIgniter framework has four main parts in the default URL pattern. First, we have the name of the server, and next, we have the name of the controller class followed by name of the controller function and function parameters at the end. Codeigniter is accessed using the URL helper. The basic URL structure is:

```
http://servername/controllerName/controllerFunction/parameter1/parameter2/.../parameter
```

- **Example:**

```
interviewbit.com/user/edit/suresh
```

Here, `interviewbit.com` is a server name, a `user` is a controller class that needs to be invoked, an `edit` is an action or method, and `suresh` is an optional action parameter that is passed to controllers.

25. How you can add or load a model in CodeIgniter?

- In CodeIgniter, models are loaded as well as called inside your controller methods. For loading a model, you must use the below-given method:

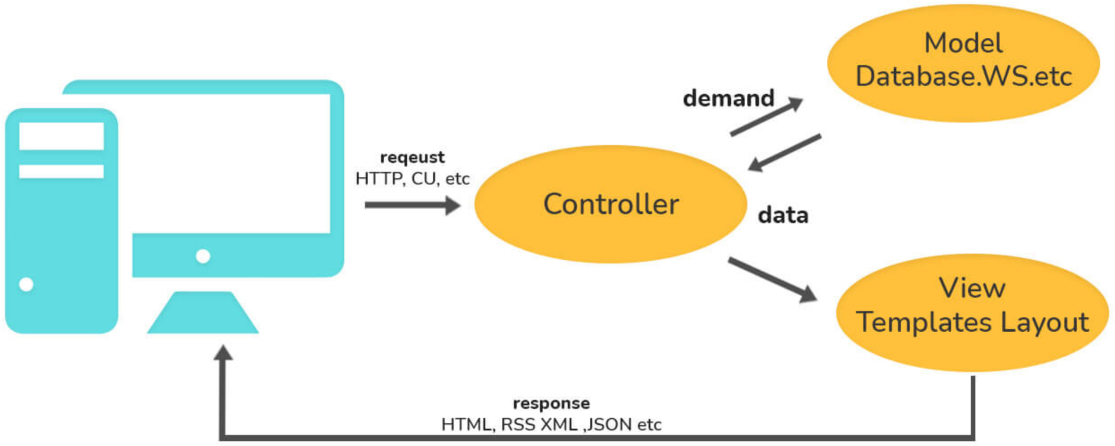
```
$this->load->model('name_of_the_model');
```

- Include the relative path from the directory of your model, if your model is placed inside a sub-directory. Consider an example, you have a model which is placed at `application/models/blog/AllPosts.php` you can load it by using: `$this->load->model('blog/AllPosts');`
- You can access the methods provided by the model, once the model gets loaded by using an object which has the same name as your controller:

```
class MyBlogController extends CI_Controller
{
   public function MyblogModel()
   {
           $this->load->model('blog_model');
           $data['que'] = $this->blog_model->get_last_five_entries();
           $this->load->view('blog_model', $data);
   }
}
```

26. Explain the CodeIgniter framework.

The CodeIgniter application is based on the MVC (Model – View – Controller) model, which separates the application logic from the presentation view. Because of presentation view separation from the PHP scripting, it allows your web pages for script minimization.



CodeIgniter MVC Framework

- **Model:**
  - Generally, a model is used for database interaction. When a user raises a request for the specific data from the application, the model takes the accountability to fetch out the records from the database table.
  - Also, a data structure represented by the model can be used to perform several operations like retrieve, insert, update, and delete.
- **Controller:**
  - The working of the CodeIgniter application is controlled by the controller. It acts as an intermediary for the communication between the model and the view. Therefore, it has the accountability to receive the user request and handle that request by furnishing a result generated by the model. The appropriate records will be displayed to the user by using the view component. (Note: The Controller file name and class name should be the same and must be in uppercase letters. Example- Main.php)
- **View:**
  - Typically, a view is similar to a web page that has the information displayed to the user. A view can also be an integral part of a web page such as header and footer. The view page can be represented in both RSS(RDF Site Summary) and a user interface.

27. How to pass an array from the controller to view in CodeIgniter?

A view is a webpage that shows each element of the user interface. It cannot be called directly, you need to load the views via the controller. You can pass an array from the controller to view in CodeIgniter using below given steps:

- **Create a view:**

Create a new text file and name it `ciblogview.php`. Save the created file in the `application/views/` directory. Open the `ciblogview.php` file and add the below-given code to it:

```
<html>
<head>
   <title>Blog</title>
</head>
<body>
   <h1>Welcome to Blog in CodeIgniter</h1>
</body>
</html>
```

- **Load the view:**

Loading a view is executed using the following syntax: `$this->load->view('name');`

Where ‘name’ represents the name of the view.

The below code creates a controller named `Blog.php` . This controller has the method for loading the view.

```
<?php
class Blog extends CI_Controller
{
   public function index()
   {
       $this->load->view('ciblogview');
   }
}
?>
```

- **Passing an array from the controller to view:**

You are allowed to paste the below-given controller code within your controller file or put it in the controller object.

```
$data['mega_header'][] = (object) array('title' => 'image portfolio' , 'img' => 'https:
$this->load->view('multiple_array', $data);
```

Arrays are displayed as a brick[‘…’] and objects as an arrow(->). You are allowed to access an array with the brick[‘…’] and object using the arrow (->). Therefore, add the below-given code in the view file:

```
<?php
   if (isset($mega_header)){
       foreach ($mega_header as $key) {
           ?>
           <div class="header_item">
               <img alt="<?php echo($key['title']); ?>" src="<?php echo($key->img); ?>'
           </div>
           <?php
       }
   }
?>
```

- **Add dynamic data to views:**

Usually, data transfer from the controller to view is done through an array or an object. The array or the object is passed as the second parameter of the view load method similar to the below-given method:

```
$data = array(
   'title' => 'TitleValue',
   'heading' => 'HeadingValue'
);
$this->load->view('ciblogview', $data);
```

The controller will look like this:

```
<?php
   class Blog extends CI_Controller {
   public function index()
   {
       $data['title'] = "TitleValue";
       $data['heading'] = "HeadingValue";
       $this->load->view('ciblogview', $data);
   }
}
?>
```

The view file will look like this:

```
<html>
<head>
   <title><?php echo $title;?></title>
</head>
<body>
   <h1><?php echo $heading;?></h1>
</body>
</html>
```

28. Explain how to prevent CodeIgniter from CSRF(Cross Site Request Forgery).

There are many ways to protect CodeIgniter from CSRF, one method of doing this is to use a hidden field in every form on the website. This hidden field is considered as CSRF token, it is a random value that changes with each HTTP request sent. After gets inserted into the website forms, it will be saved in the user's session as well. So, when the user submits the form, the website checks whether it is the same as the one that was saved in the session. If it is the same then, the request is authorized.

29. What are the sessions in CodeIgniter? How to handle sessions in CodeIgniter?

In CodeIgniter, you are allowed to maintain a user's "state" by Session class and keep an eye on their activity while they browse your website.

- **Loading a session in CodeIgniter:**
  For using session, your controller should be loaded with your Session class by using the `$this->load->library(‘session’);` .
  Once the Session class is loaded, the Session library object can be obtained using `$this->session` .
- **Read session data in CodeIgniter:**
  `$this->session->userdata();` method of Session class is used to read or obtain session data in CodeIgniter.
  Usage: `$this->session->userdata('name_of_user');`
  Also, the below-given method of the Session class can be used to read session data.
  Usage: `$this->session->key_item`
  Where an item represents the key name you want to access.
- **Create a session in CodeIgniter:**
  The set_userdata() method that belongs to the Session class is useful in creating a session in CodeIgniter. This method uses an associative array that has the data you want to include in the session.
  Adding session data:
  Example:

```
$sessiondata = array(
   'name_of_user'  => 'lekha',
   'email'     => 'lekha@interviewbit.com',
   'log_state' => TRUE
);
$this->session->set_userdata($sessiondata);
```

If you want to add a single user data at a time, `set_userdata()` supports this syntax:

```
$this->session->set_userdata('demo_username', 'demo_value');
```

- **Remove session data in CodeIgniter:**
  The unset_userdata() method that belongs to the Session class is useful for removing session data in CodeIgniter. Usage examples are given below:
  Unset particular key:

```
$this->session->unset_userdata('name_of_user');
```

Unset an array of item keys:

```
$arr_items = array('name_of_user', 'email');
$this->session->unset_userdata($arr_items);
```

30. Explain CodeIgniter folder structure.

The CodeIgniter folder structure is given below:

- **application:** This directory will have your application logic. All of your application codes will be held in this directory. Internal subdirectories in the CodeIgniter directory structure are given below:
    - cache – It stores cached files.
    - config – It keeps configuration files.
    - controller – All application controllers are defined under this controller.
    - core – It consists of custom core classes that extend system files. For example, if you create a base controller that other controllers should extend, then you should place it under this directory.
    - helpers – This directory will be used for user-defined helper functions.
    - hooks – It is used for custom hooks in the CodeIgniter folder structure.
    - language – It is used to store language files for applications that use multiple languages.
    - libraries – It is used to store custom-created libraries.
    - logs – Application log files are placed in this directory.
    - models - All application models must be defined under this directory.
    - third_party – This is used for custom many packages that are created by you or other developers.
    - views – application views will be stored in this directory.
- **system:** It consists of the framework core files. It is not advised to make any modifications in this directory or put your own application code into this directory. System subdirectories in CodeIgniter are given below:
    - core – This is considered to be the heart of the CodeIgniter Framework. All of the core files that construct the framework are located here. If you would like to extend the core file functionality, then you must
    - create a custom core file in the application directory. After this, you are allowed to override or add new behavior that you wish. You should never make any changes directly in this directory.
    - database – It stores the files such as database drivers, cache, and other files that are needed for database operations.
    - fonts – This directory contains fonts and font-related information.
    - helpers – This directory consists of helper functions that come out of the box.
    - language – It contains language files that are used by the framework
    - libraries – It contains the source files for the different libraries that come along with CodeIgniter out of the box.
- **user_guide:** This directory consists of a user manual for CodeIgniter. You should

31. What is the security parameter for XSS in CodeIgniter?

- Codeigniter has got a Cross-Site Scripting(XSS) hack prevention filter. This filter either automatically runs or you can run it based on item, to filter all data related to POST and COOKIE.
- The XSS filter will target the frequently used methods to trigger JavaScript code or other types of code that attempt to hijack cookies or do any other malicious activity. If it identifies anything suspicious or anything disallowed is encountered, then it will convert the data to character entities.
- To filter data through the XSS filter, we will make use of the xss_clean() method as given below:

```
$data = $this->security->xss_clean($data);
```

This function is used only when you are submitting data. The second Boolean parameter is optional and used to check the image files for the XSS attacks. This is very useful for file upload. If its value is true, that means the image is safer and not otherwise.

32. What is the default controller in CodeIgniter?

- When the name of the file is not mentioned in the URL then the file will be specified in the default controller that is loaded by default. By default, the file name will be `welcome.php` , which is known as the first page to be seen after the installation of CodeIgniter.
- `localhost/codeigniter/` In this case, the `welcome.php` will be generally loaded as the file name is not mentioned in the provided URL. Generally, the programmers can change the default controller that is present in the `application/config/routes.php` file as per their needs.
- `$route['default_controller'] = ' ';` In the above-given syntax, the programmer has to specify the file name that he/she wants to get loaded as the default one.

33. List all the auto-loadable resources available in CodeIgniter.

- The below-given items can be automatically loaded in CodeIgniter:
  - Classes obtained in the directory named `libraries/`
  - Custom config files obtained in the directory named `config/`
  - Helper files obtained in the directory named `helpers/`
  - Models obtained in the directory named `models/`
  - Language files obtained in the directory named `system/language/`
- For resource autoloading, you should open the file `application/config/autoload.php` and include the item that you want to be get loaded into the array of autoloads. In the file related to each type of item, you can find instructions.

34. What do you mean by the controller in CodeIgniter?

- The mediator present between the model and the view for processing the HTTP request and is used for generating a web page is called a controller. It is considered as the center of each HTTP request that exists on the web application of the user.
- Consider the following URL in this reference: `projectName/index.php/welcome/` In this URL, the CodeIgniter is trying to find the `welcome.php` file and the Welcome class.
- Controller syntax is given below:

```
class ControllerName extends CI_Controller
{
   public function __construct()
   {
       parent::__construct();
   }
   public function MethodName()
   {
   }
}
```

35. Why is there a need to configure the URL routes?

Changing the URL routes has many benefits such as:

- From the SEO(Search Engine Optimization) point of the view, to make URL SEO friendly and obtain more user visits.
- Hide some URL elements like controller name, function name, etc. from the users for security purposes.
- Provides different functionality to the specific parts of a system.

Conclusion:

CodeIgniter has become the predominant choice of interest for developers while creating websites. Even though there are plenty of other choices when considering the right programming framework for the applications, CodeIgniter has gained its own preference in the market. The sole reason that most of the developers are choosing it is because of the several advantages that it offers to the people.

Freedcamp, Nissan, Bonfire, Buffer, etc. are some of the websites that use the CodeIgniter framework. There is a great amount of demand for PHP developers with CodeIgniter skills.

This set of interview questions will definitely help you excel in the CodeIgniter job interview. We hope these interview questions and answers on CodeIgniter will assist you to crack the relevant interview.

References:

“Professional CodeIgniter” by Thomas Myer.

“CodeIgniter for Rapid PHP Application Development” by David Upton.

Codeigniter Guide

Codeigniter Download

Links to More Interview Questions

- C Interview Questions
- Web Api Interview Questions
- Cpp Interview Questions
- Machine Learning Interview Questions
- Css Interview Questions
- Django Interview Questions
- Operating System Interview Questions
- Git Interview Questions
- Dbms Interview Questions
- Pl Sql Interview Questions
- Ansible Interview Questions
- Php Interview Questions
- Hibernate Interview Questions
- Oops Interview Questions
- Docker Interview Questions
- Laravel Interview Questions
- Dot Net Interview Questions
- React Native Interview Questions
- Java 8 Interview Questions
- Spring Boot Interview Questions
- Tableau Interview Questions
- Java Interview Questions
- C Sharp Interview Questions
- Node Js Interview Questions
- Devops Interview Questions
- Mysql Interview Questions
- Asp Net Interview Questions
- Kubernetes Interview Questions
- Aws Interview Questions
- Mongodb Interview Questions
- Power Bi Interview Questions
- Linux Interview Questions
- Jenkins Interview Questions