**MVC VS Framework**

**Design pattern- means manage code in module wise**

**Type of design pattern-**

1>mvc

2>dependency injection

3>decoration pattern

4>Factory pattern

5>singleton pattern

**Mvc – mvc is a design pattern**

**Framework-collection of reusability library**

**Question: What is CodeIgniter?**  
Codeigniter is an open source framework for web application on PHP. It is loosely based on MVC pattern and it is similar to CakePHP.

**Question: What is Stable version of CodeIgniter?**  
**Version:** 3.1.9,   
**Date** Jun 12, 2018

**Question: In which language CodeIgniter is written?**  
PHP

**Question: What are the features of codeigniter?**

Open source framework  
Light Weight  
CodeIgniter is Extensible  
Full Featured database classes

Easy to understand

Documentation is very good **Question: How to access config variable in codeigniter?**

$this->config->item('variable name'); **Question: How to unset session in codeigniter?**

$this->session->unsetuserdata('somename');; **Question: How do you get last insert id in codeigniter?**

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

**Question: How to print SQL statement in codeigniter model??**

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

**Question: What are hooks in CodeIgniter?**  
CodeIgniter's Hooks feature provides a means to tap into and modify the inner workings of the framework without hacking the core files.   
 $hook['pre\_controller'] = array(

'class' => 'MyClass',

'function' => 'Myfunction',

'filename' => 'Myclass.php',

'filepath' => 'hooks',

'params' => array('param1', 'param2', 'param3')

);

**Question: How to load model in CodeIgniter?**

$this->load->model ('Model\_Name');  
**Question: What is offical Website URL?**  
[http://www.codeigniter.com](http://www.codeigniter.com/)  
  
**Question: List out different types of hook in Codeigniter?**

* post\_controller\_constructor
* pre\_controller
* pre\_sytem
* post\_sytem
* cache\_override
* display\_override
* post\_controller

**4. Explain codeigniter file structure.**  
**Application**  
-cache  
-config  
-controllers

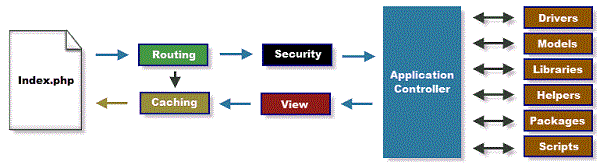
-core  
-errors  
-helpers  
-hooks  
-languages

-Libraries  
-logs  
-models-  
-thirdparty  
-view  
**system**  
-core  
-database  
-fonts  
-helpers  
-language  
-libraries  
 **what is mvc framework?**  
 **Controller**: The controller serves as an intermediary between the model and view.  
**Model**: The model represents your data structures. The model consists of application data and business rules.  
**View**: the view can be any output representation of data.

**4. What are the features of Codeigniter?**

1. Codeigniter is free to use, it’s an open source framework.  
2. Its light weight. The core system requires only a few very small libraries. Not like other frameworks that require heavy file libraries.  
3. Codeigniter is fast. It’s faster than any other framework in php.  
4. The URLs generated by Codeigniter are clean and search-engine friendly. You will change any url to whatever you want from files.  
5. Codeigniter is Extensible. The system can be easily extended through the use of your own libraries, helpers, or through class extensions or system hooks.  
6. Codeigniter Uses MVC(Model View Controller) which allows great separation between logic and presentation.  
7. Codeigniter requires nearly zero configuration, does not require you to use the command line, not forced to learn a templating language.  
8. Full Featured database classes with support for several platforms, Security and XSS Filtering, Error Logging.

**6. Explain Application Flow Chart in codeigniter.**



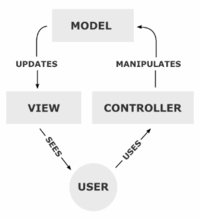
Application flow chart from Codeigniter documentation:

1. The index.php serves as the front controller, initializing the base resources needed to run CodeIgniter.  
2. The Router examines the HTTP request to determine what should be done with it.  
3. If a cache file exists, it is sent directly to the browser, bypassing the normal system execution.  
4. Security. Before the application controller is loaded, the HTTP request and any user submitted data is filtered for security.  
5. The Controller loads the model, core libraries, helpers, and any other resources needed to process the specific request.  
6. The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served.

**7. Explain MVC in Codeigniter.**

Model–View–Controller (MVC) is an architecture that separates the representation of information from the user’s interaction with it.

**Controller: -** The **Controller** serves as an *intermediary* between the Models, the View. *Controller* mediates input, converting it to commands for the model or view.



**Model:-**The **Model** represents your data structures. Typically your model classes will contain functions that help you retrieve, insert, and update information in your database. The *model* consists of application data and business rules.  
**View:-**The **View** is the information that is being presented to a user. A View will normally be a web page. A *view* can be any output representation of data.  
For more detail understanding MVC please read this article what is MVC (Model-View-Controller) Architecture.

**9. How you will add or load an model in codeigniter.**

Models will typically be loaded and called from within your controller functions. To load a model you will use the following function:

|  |  |
| --- | --- |
|  | **$this->load-model('Model\_name')**; |

**10. What are the helpers in codeigniter.**

Helpers, as the name suggests, help you with tasks. Each helper file is simply a collection of functions in a particular category. There are URL Helpers that assist in creating links, there are Form Helpers that help you create form elements, Text Helpers perform various text formatting routines, Cookie Helpers set and read cookies, File Helpers help you deal with files, etc.  
Loading a helper file is quite simple using the following function:

|  |  |
| --- | --- |
|  | **$this->load->helper('name');** |

**11. How you will use or add Codeigniter libraries.**

All of the available libraries are located in your system/libraries folder. In most cases, to use one of these classes involves initializing it within a controller using the following initialization function:-

|  |  |
| --- | --- |
|  | **$this->load->library('class name')**; |

**12. How you will work with error handling in Codeigniter.**

Codeigniter lets you build error reporting into your applications using the functions:-  
  
**1. show\_error():-** This function will display the error message supplied to it using template application/errors/error\_general.php.  
**2. show\_404() :-** Function will display the 404 error message.  
**3.** **log\_message(‘level’, ‘message’) :-** This function lets you write messages to your log files. You must supply one of three “levels” in the first parameter, indicating what type of message it is (debug, error, info), with the message itself in the second parameter.  
  
**13. What are the security parameters for XSS in CodeIgniter?**  
  
Codeigniter has got a **cross(X)-site scripting** hack prevention filter. This filter either runs automatically or you can run it as per item basis, to filter all POST and COOKIE data that come across. The XSS filter will target the commonly used methods to trigger JavaScript or other types of code that attempt to hijack cookies or other malicious activity. If it detects any suspicious thing or anything disallowed is encountered, it will convert the data to character entities.   
  
**14**. **Explain how you can enable CSRF (Cross Site Request Forgery) in Codeigniter?**   
  
You can activate CSRF (Cross Site Request Forgery) protection in Codeigniter by operating your application/config/config.php file and setting it to  
$config [ ‘csrf\_protection’] = TRUE;  
If you avail the form helper, the **form\_open()** function will insert a hidden csrf field in your forms automatically.  
  
**15. Explain how you can prevent Codeigniter from CSRF?**  
  
There are several ways to protect Codeigniter from CSRF, one way of doing is to use a hidden field in each form on the website.  This hidden field is referred as CSRF token; it is nothing but a random value that alters with each HTTP request sent. As soon as it is inserted in the website forms, it gets saved in the user’s session as well.  So, when the form is submitted by the users, the website checks whether it is the same as the one saved in the session. If it is same then, the request is legitimate.   
  
**16. Explain what is inhibitor in Codeigniter?**  
  
For Codeigniter, **inhibitor** is an error handler class, using the native PHP functions like set\_exception\_handler, set\_error\_handler, register\_shutdown\_function to handle parse errors, exceptions, and fatal errors.   
  
**17. Explain routing in Codeigniter?**  
  
In Codeigniter, the way of PHP files served is different rather than accessing it directly from the browser. This process is called routing. **Routing in Codeigniter gives you freedom to customize the default URL pattern to use our own URL pattern according to the requirement**. So, whenever there is a request made and matches our URL pattern it will automatically direct to the specified controller and function.   
  
**18. Why is there a need to configure the URL routes?**  
  
Changing the URL routes has some benefits like

* From SEO point of view, to make URL SEO friendly and get more user visits
* Hide some URL element such as a function name, controller name, etc. from the users for security reasons
* Provide different functionality to particular parts of a system.
* **13. How do you use aliases with auto loading models in Codeigniter?**

We can auto load model like this:

$autoload['model'] = array(array('usersmodel', 'users'), array('news*model', 'news'), 'category*model');

**14. How to get random records in mysql using codeigniter?**

We can use this:

$this->db->order*\_by('id','RANDOM');*

Or

$this->db->order\_by('rand()');

**15. How to print SQL statement in codeigniter model?**

We can use this:

$this->db->last\_*query();*

*Or* print\_r($query)

You can also use (in view) and you will see the SQL statement:

$this->output->enable\_profiler(TRUE);

**17. What is the default URL pattern used in Codeigniter framework?**

Codeigniter **framework** **URL** **has four main components in default URL pattern**. First we have the **server** **name** and next we have the **controller class** **name** followed by **controller function** **name** and **function parameters** at the end. Codeigniter can be accessed using the URL helper. For example <http://servername/controllerName/controllerFunction/parameter1/parameter2>.

**18. How do you get last insert id in codeigniter ?**

We can get with this.

$this->db->insert\_*id();*

**19. How do you set default timezone in codeigniter?**

We can do by adding date*\_default\_*timezone*\_set('America/Los*Angeles');

in index.php

**12) Explain how you can extend the class in Codeigniter?**

To extend the native input class in Codeigniter, you have to build a file named application/core/**MY\_Input.php** and declare your class with

Class **MY\_Input** extends CI\_Input {}