1. What is a Web Server

Web server is a computer where the web content is stored. Basically a web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

2. What do the initials of PHP stand for?

PHP: Hypertext Preprocessor

3. Which programming language does PHP resemble?

Perl and C

4. What does PEAR stand for?

PHP Extension and Application Repository it extends PHP and provides a higher level of programming for web developers.

5. What is the actually used PHP version?

PHP version 7.4 is the most used version.

6. How do you execute a PHP script from the command line?

Just use the PHP command line interface (CLI) and specify the file name of the script to be executed as follows:

php script.php

7. How to run the interactive PHP shell from the command line interface?

Just use the PHP CLI program with the option -a as follows:

php -a

8.What is the correct and the most two common way to start and finish a PHP block of code?

<?php [ --- PHP code---- ] ?> and <? [--- PHP code ---] ?>

9.How can we display the output directly to the browser?

we have to use the special tags <? = and ?>.

10.What is the main difference between PHP 4 and PHP 5?

PHP 5 presents many additional OOP (Object Oriented Programming) features.

11.What is OOPs in PHP ?

OOP stands for Object-Oriented Programming.

Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.

Object-oriented programming has several advantages over procedural programming:

* OOP is faster and easier to execute
* OOP provides a clear structure for the programs
* OOP helps to keep the PHP code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
* OOP makes it possible to create full reusable applications with less code and shorter development time.

12.What is the use of Object ?

Object is a compound data type (along with arrays). Values of more than one types can be stored together in a single variable. Object is an instance of either a built-in or user defined class. In addition to properties, class defines functionality associated with data.

13. What is different between constructor and destructor?

constructor : A constructor allows you to initialize an object's properties upon creation of the object.If you create a \_\_construct() function, PHP will automatically call this function when you create an object from a class.

Notice that the construct function starts with two underscores (\_\_)!

Destructor: A destructor is called when the object is destructed or the script is stopped or exited.

If you create a \_\_destruct() function, PHP will automatically call this function at the end of the script.

Notice that the destruct function starts with two underscores (\_\_)!

14. What is Abstract Class ?

-> Abstract classes and methods are when the parent class has a named method, but need its child class(es) to fill out the tasks.

An abstract class is a class that contains at least one abstract method. An abstract method is a method that is declared, but not implemented in the code.

An abstract class or method is defined with the abstract keyword

15. Is multiple inheritance supported in PHP?

-> PHP supports only single inheritance; it means that a class can be extended from only one single class using the keyword ‘extended’.

16. What is the meaning of a final class and a final method?

-> ‘final’ is introduced in PHP5. Final class means that this class cannot be extended and a final method cannot be overridden.

17. How is the comparison of objects done in PHP?

-> We use the operator ‘==’ to test is two objects are instanced from the same class and have same attributes and equal values. We can test if two objects are referring to the same instance of the same class by the use of the identity operator ‘===’.

18. How can PHP and HTML interact?

-> It is possible to generate HTML through PHP scripts, and it is possible to pass pieces of information from HTML to PHP.

19. What type of operation is needed when passing values through a form or an URL?

->

If we would like to pass values through a form or an URL, then we need to encode and to decode them using htmlspecialchars() and urlencode().

20. How can PHP and Javascript interact?

-> PHP and Javascript cannot directly interact since PHP is a server side language and Javascript is a client-side language. However, we can exchange variables since PHP can generate Javascript code to be executed by the browser and it is possible to pass specific variables back to PHP via the URL.

21. What is needed to be able to use image function?

-> GD library is needed to execute image functions.

22. What is the use of the function 'imagetypes()'?

-> imagetypes gives the image format and types supported by the current version of GD-PHP

23. What are the functions to be used to get the image's properties (size, width, and height)?

-> The functions are getimagesize() for size, imagesx() for width and imagesy() for height.

24. How failures in execution are handled with include() and require() functions?

-> If the function require() cannot access to the file then it ends with a fatal error. However, the include() function gives a warning and the PHP script continues to execute.

25. What is the main difference between require() and require\_once()?

-> require(), and require\_once() perform the same task except that the second function checks if the PHP script is already included or not before executing it.

(same for include\_once() and include())

26. How can I display text with a PHP script?

-> Two methods are possible:

<!--?php echo "Method 1"; print "Method 2"; ?-->

27. How can we display information of a variable and readable by a human with PHP?

-> To be able to display a human-readable result we use print\_r().

28. How is it possible to set an infinite execution time for PHP script?

->The set\_time\_limit(0) added at the beginning of a script sets to infinite the time of execution to not have the PHP error ‘maximum execution time exceeded.’ It is also possible to specify this in the php.ini file.

29. What does the PHP error 'Parse error in PHP - unexpected T\_variable at line x' means?

-> This is a PHP syntax error expressing that a mistake at the line x stops parsing and executing the program.

30. What should we do to be able to export data into an Excel file?

-> The most common and used way is to get data into a format supported by Excel. For example, it is possible to write a .csv file, to choose for example comma as a separator between fields and then to open the file with Excel

31. What is the function file\_get\_contents() useful for?

-> The file\_get\_contents() function in PHP is an inbuilt function that is used to read a file and storing it is a string variable.

32. How can we connect to a MySQL database from a PHP script?

-> To be able to connect to a MySQL database, we must use mysqli\_connect() function as follows:

<!--?php $database = mysqli\_connect("HOST", "USER\_NAME", "PASSWORD"); mysqli\_select\_db($database,"DATABASE\_NAME"); ?-->

33. What is the function mysql\_pconnect() useful for?

-> mysql\_pconnect() ensure a persistent connection to the database, it means that the connection does not close when the PHP script ends.

This function is not supported in PHP 7.0 and above

34. How be the result set of Mysql handled in PHP?

-> The result set can be handled using mysql\_fetch\_array, mysql\_fetch\_assoc, mysql\_fetch\_object or mysql\_fetch\_row.

35. How is it possible to know the number of rows returned in the result set?

-> The mysqli\_num\_rows() function returns the number of rows in a result set.

36. Which function gives us the number of affected entries by a query?

-> The affected\_rows / mysqli\_affected\_rows() function returns the number of entries affected by an SQL query

37. What is the difference between mysqli\_fetch\_object() and mysqli\_fetch\_array()?

-> The mysqli\_fetch\_object() function collects the first single matching record where mysqli\_fetch\_array() collects all matching records from the table in an array.

38. How can we access the data sent through the URL with the GET method?

-> To access the data sent via the GET method, we use $\_GET array like this:

www.url.com?var=value

$variable = $\_GET["var"]; this will now contain 'value'

39. How can we access the data sent through the URL with the POST method?

-> To access the data sent this way, you use the $\_POST array.

Imagine you have a form field called ‘var’ on the form when the user clicks submit to the post form, you can then access the value like this:

$\_POST["var"];

40. How can we check the value of a given variable is a number?

-> It is possible to use the dedicated function, is\_numeric() to check whether it is a number or not

41. How can we check the value of a given variable is alphanumeric?

-> It is possible to use the dedicated function, ctype\_alnum to check whether it is an alphanumeric value or not.ctype\_alnum (Check for Alphanumeric)

42. How do I check if a given variable is empty?

-> If we want to check whether a variable has a value or not, it is possible to use the empty() function.

43. What does the unlink() function mean?

-> The unlink() function is dedicated for file system handling. It simply deletes the file given as entry.

44. What does the unset() function mean?

-> The unset() function is dedicated for variable management. It will make a variable undefined.

45. How do I escape data before storing it in the database?

-> The addslashes function enables us to escape data before storage into the database.

46. How is it possible to remove escape characters from a string?

-> The stripslashes function enables us to remove the escape characters before apostrophes in a string.

47. How can we automatically escape incoming data?

-> We have to enable the Magic quotes entry in the configuration file of PHP.

48. What does the function get\_magic\_quotes\_gpc() means?

-> The function get\_magic\_quotes\_gpc() tells us whether the magic quotes is switched on or no.

49. Is it possible to remove the HTML tags from data?

-> The strip\_tags() function enables us to clean a string from the HTML tags.

50. what is the static variable in function useful for?

-> A static variable is defined within a function only the first time, and its value can be modified during function calls as follows:

<!--?php function testFunction() { static $testVariable = 1; echo $testVariable; $testVariable++; } testFunction(); //1 testFunction(); //2 testFunction(); //3 ?-->

51. can we define a variable accessible in functions of a PHP script?

-> This feature is possible using the global keyword.

52. How is it possible to return a value from a function?

-> A function returns a value using the instruction ‘return $value;’.

53. What is the most convenient hashing method to be used to hash passwords?

-> hash passwords to secure our passwords while creating websites and storing our database.used like md5, crypt, sha1, and bcrypt. And the most commonly used nowadays is bcrypt hashing method

54. Which cryptographic extension provide generation and verification of digital signatures?

-> The PHP-OpenSSL extension provides several cryptographic operations including generation and verification of digital signatures.

55. How is a constant defined in a PHP script?

-> The define() directive lets us defining a constant as follows:

define ("ACONSTANT", 123);

56. How can you pass a variable by reference?

-> To be able to pass a variable by reference, we use an ampersand in front of it, as follows $var1 = &$var2

57. Will a comparison of an integer 12 and a string "13" work in PHP?

-> “13” and 12 can be compared in PHP since it casts everything to the integer type.

58. How is it possible to cast types in PHP?

-> The name of the output type has to be specified in parentheses before the variable which is to be cast as follows:

\* (int), (integer) – cast to integer

\* (bool), (boolean) – cast to boolean

\* (float), (double), (real) – cast to float

\* (string) – cast to string

\* (array) – cast to array

\* (object) – cast to object

59. When is a conditional statement ended with endif?

-> Endif is a statement to end an if conditional using the alternate syntax. Rather than:

<?php

if (!alternateSyntax) {

// normal conditional syntax has open

// and closing curly braces

}

?>

60. How is the ternary conditional operator used in PHP?

-> It is composed of three expressions: a condition, and two operands describing what instruction should be performed when the specified condition is true or false as follows:

Expression\_1?Expression\_2 : Expression\_3;

61. What is the function func\_num\_args() used for?

-> The function func\_num\_args() is used to give the number of parameters passed into a function.

62. If the variable $var1 is set to 10 and the $var2 is set to the character var1, what's the value of $$var2?

-> $var1=10

$var2 is char var1

$$var2 contains the value 10.

63. What does accessing a class via :: means?

-> :: is used to access static methods that do not require object initialization.

64. In PHP, objects are they passed by value or by reference?

-> In PHP, objects are passed by references by default.

65. Are Parent constructors called implicitly inside a class constructor?

-> No, a parent constructor have to be called explicitly as follows:

parent::constructor($value)

66. What's the difference between \_\_sleep and \_\_wakeup?

-> \_\_sleep : It is used to return the array of all the variables which need to be saved.

magic methods.

It is executed before the serialize() command.

\_\_wakeup : It is used to retrieve all the arrays returned by the \_sleep() command.

magic methods

Is executed before the unserialize() command.

67. What is faster?

-> 1- Combining two variables as follows:

$variable1 = 'Hello ';

$variable2 = 'World';

$variable3 = $variable1.$variable2;

68. what is the definition of a session?

-> A session is a way to store information (in variables) to be used across multiple pages.

Unlike a cookie, the information is not stored on the users computer.

PHP global variable: $\_SESSION

69. How to initiate a session in PHP?

-> Start a PHP Session

A session is started with the session\_start() function. Session variables are set with the PHP global variable: $\_SESSION.

70. How can you propagate a session id?

-> There are two methods to propagate a session id: Cookies. Or URL parameter.

71. What is the meaning of a Persistent Cookie?

-> A persistent cookie is permanently stored in a cookie file on the browser’s computer. By default, cookies are temporary and are erased if we close the browser.

72. When do sessions end?

-> Sessions automatically end when the PHP script finishes executing but can be manually ended using the session\_write\_close().

73. Explain MVC in PHP

-> CodeIgniter is based on the Model-View-Controller (MVC) development pattern. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting.

74. What is the difference between session\_unregister() and session\_unset()?

-> The session\_unregister() function unregister a global variable from the current session .and

the session\_unset() function frees all session variables

75. What does $GLOBALS mean?

-> $GLOBALS is associative array including references to all variables which are currently defined in the global scope of the script.

76. What does $\_SERVER mean?

-> $\_SERVER is an array including information created by the web server such as paths, headers, and script locations.

77. What does $\_FILES means?

-> $\_FILES is an associative array composed of items sent to the current script via the HTTP POST method.

78. What is the difference between $\_FILES['userfile']['name'] and $\_FILES['userfile']['tmp\_name']?

-> $\_FILES[‘userfile’][‘name’] represents the original name of the file on the client machine,

$\_FILES[‘userfile’][‘tmp\_name’] represents the temporary filename of the file stored on the server.

79. How can we get the error when there is a problem to upload a file?

-> $\_FILES[‘userfile’][‘error’] contains the error code associated with the uploaded file.

80. How can we change the maximum size of the files to be uploaded?

-> We can change the maximum size of files to be uploaded by changing upload\_max\_filesize in php.ini.

81. What does $\_ENV mean?

-> $\_ENV is an associative array of variables sent to the current PHP script via the environment variables.

82. What does $\_COOKIE mean?

-> $\_COOKIE is an associative array of variables sent to the current PHP script using the HTTP Cookies.

83. What does the scope of variables mean?

-> The scope of a variable is the context within which it is defined. For the most part, all PHP variables only have a single scope. This single scope spans included and required files as well.

84. what the difference between the 'BITWISE AND' operator and the 'LOGICAL AND' operator?

-> The logical AND operator works on Boolean expressions, and returns Boolean values only.

The bitwise AND operator works on integer, short int, long, unsigned int type data, and also returns that type of data.

85. What are the two main string operators?

-> The first is the concatenation operator (‘.’), which returns the concatenation of its right and left arguments. The second is (‘.=’), which appends the argument on the right to the argument on the left.

86. What does the array operator '===' means?

-> $a === $b TRUE if $a and $b have the same key/value pairs in the same order and of the same types.

87. What is the differences between $a != $b and $a !== $b?

-> != means inequality (TRUE if $a is not equal to $b) and !== means non-identity (TRUE if $a is not identical to $b).

88. How can we determine whether a PHP variable is an instantiated object of a certain class?

-> To be able to verify whether a PHP variable is an instantiated object of a certain class we use instanceof.

89. What is the goto statement useful for?

-> The goto statement can be placed to enable jumping inside the PHP program. The target is pointed by a label followed by a colon, and the instruction is specified as a goto statement followed by the desired target label.

90. what is the difference between Exception::getMessage and

-> Exception::getMessage : This function returns the exception message.

It returns the exception message in string format.

It is helpful in all types of codes.

Exception::getLine

lets us getting the line in which the exception occurred.

91. Exception:: getLine?

-> Exception::getLine :

This function returns the position of the line at which the exception has occurred

It returns the line number in integer format.

It is most helpful in huge codes ie. Codes containing many lines.

92. What does the expression Exception::\_\_toString means?

-> Exception::\_\_toString

gives the String representation of the exception

93. How is it possible to parse a configuration file?

-> The function parse\_ini\_file() enables us to load in the ini file specified in filename and returns the settings in it in an associative array.

94. How can we determine whether a variable is set?

-> The boolean function isset determines if a variable is set and is not NULL.

95. What is the difference between the functions strstr() and stristr()?

-> The string function strstr(string allString, string occ) returns part of allString from the first occurrence of occ to the end of allString. This function is case-sensitive. stristr() is identical to strstr() except that it is case insensitive.

96. what is the difference between for and foreach?

-> for (expr1; expr2; expr3)

statement

The first expression is executed once at the beginning. In each iteration, expr2 is evaluated. If it is TRUE, the loop continues, and the statements inside for are executed. If it evaluates to FALSE, the execution of the loop ends. expr3 is tested at the end of each iteration.

However, foreach provides an easy way to iterate over arrays, and it is only used with arrays and objects.

97. Is it possible to submit a form with a dedicated button?

-> It is possible to use the document.form.submit() function to submit the form. For example: <input type=button value="SUBMIT" onClick="document.form.submit()">

98. What is the difference between ereg\_replace() and eregi\_replace()?

-> The function eregi\_replace() is identical to the function ereg\_replace() except that it ignores case distinction when matching alphabetic characters.

99. Is it possible to protect special characters in a query string?

-> Yes, we use the urlencode() function to be able to protect special characters.

100. What are the three classes of errors that can occur in PHP?

-> The three basic classes of errors are notices (non-critical), warnings (serious errors) and fatal errors (critical errors).

101. What is the difference between characters \034 and \x34?\

-> \034 is octal 34 and \x34 is hex 34.

102. How can we pass the variable through the navigation between the pages?

-> It is possible to pass the variables between the PHP pages using sessions, cookies or hidden form fields.

103. Is it possible to extend the execution time of a PHP script?

-> The use of the set\_time\_limit(int seconds) enables us to extend the execution time of a PHP script. The default limit is 30 seconds.

104. Is it possible to destroy a cookie?

-> Yes, it is possible by setting the cookie with a past expiration time.

105. What is the default session time in PHP?

-> The default session time in php is until the closing of the browser

106. Is it possible to use COM component in PHP?

-> Yes, it’s possible to integrate (Distributed) Component Object Model components ((D)COM) in PHP scripts which is provided as a framework.

107. Explain whether it is possible to share a single instance of a Memcache between multiple PHP projects?

-> Yes, it is possible to share a single instance of Memcache between multiple projects. Memcache is a memory store space, and you can run memcache on one or more servers. You can also configure your client to speak to a particular set of instances. So, you can run two different Memcache processes on the same host and yet they are completely independent. Unless, if you have partitioned your data, then it becomes necessary to know from which instance to get the data from or to put into.

108. Explain how you can update Memcached when you make changes to PHP?

-> When PHP changes you can update Memcached by

* Clearing the Cache proactively: Clearing the cache when an insert or update is made
* Resetting the Cache: It is similar to the first method but rather than just deleting the keys and waiting for the next request for the data to refresh the cache, reset the values after the insert or update.

109. Explain what is Codeigniter?

-> CodeIgniter is an open source and powerful framework used for developing web applications on PHP. It is loosely based on MVC pattern and similar to Cake PHP. CodeIgniter contains libraries, simple interface and logical structure to access these libraries, plug-ins, helpers and some other resources which solve the complex functions of PHP more easily maintaining high performance. It simplifies the PHP code and brings out a fully interactive, dynamic website at a much shorter time.

110. What is the current version of Codeigniter?

-> CodeIgniter 4 is the latest version of the framework,

111. How to check the version of CodeIgniter framework?

-> Using system/core/CodeIgniter.php

Using php echo command: *CI\_VERSION* is build in constant in CodeIgniter framework

112. List Databases supported By Codeigniter Frameworks?

-> Database Name Drivers to support Codeigniter

MySQL MySQL (deprecated), MYSQLI and PDO drivers

Oracle oci8 and PDO drivers

Postgresql Postgre and PDO drivers

Ms sql MsSQL, Sqlsrv (version 2005 and above only) and PDO drivers

Sql lite SQLite (version 2), sqlite3 (version 3) and PDO drivers

Cubrid Cubridand PDO drivers

interbase/firebase iBase and PDO drivers

Odbc ODBC and PDO drivers

(you should know that ODBC is actually an abstraction layer)

113. List some features provided by CodeIgniter?

-> codeigniter features:

Model-View-Controller Based System

* Extremely Light Weight
* Full Featured database classes with support for several platforms.
* Query Builder Database Support
* Form and Data Validation
* Security and XSS Filtering
* Session Management
* Email Sending Class. Supports Attachments, HTML/Text email, multiple protocols (sendmail, SMTP, and Mail) and more.
* Image Manipulation Library (cropping, resizing, rotating, etc.). Supports GD, ImageMagick, and NetPBM
* File Uploading Class
* FTP Class
* Localization
* Pagination
* Data Encryption
* Benchmarking
* Full Page Caching
* Error Logging
* Application Profiling
* Calendaring Class
* User Agent Class
* Zip Encoding Class
* Template Engine Class
* Trackback Class
* XML-RPC Library
* Unit Testing Class
* Search-engine Friendly URLs
* Flexible URI Routing
* Support for Hooks and Class Extensions
* Large library of “helper” functions

114. Explain helpers in CodeIgniter and how to load helper file?

-> To use helper files, you need to load it. Once loaded it is globally available to your controller and views. They are located at two places in CodeIgniter. CodeIgniter will look first for a helper in application/helpers folder and if not found there then it will go to system/helpers folder.

115. Explain routing in CodeIgniter?

-> Routing – routing is responsible for responding to URL requests. CodeIgniter Routing matches the URL to the pre-defined routes. If not route match is found then CodeIgniter throws a page not found exception. Controllers – routes are linked to controllers. Controllers glue the models and views together.

116. What are hooks in CodeIgniter? List them?

-> In CodeIgniter, hooks are events which can be called before and after the execution of a program. It allows executing a script with specific path in the CodeIgniter execution process without modifying the core files.

There are two hook files in CodeIgniter. One is application/config/hooks.php folder and other is application /hooks folder.

117. List Common Functions in Codeigniter?

->

| Syntax | is\_php(*$version*) |
| --- | --- |
| Parameters | $version (*string*) − Version number |
| Return | TRUE if the running PHP version is at least the one specified or FALSE if not |
| Return Type | void |
| Description | Determines if the PHP version being used is greater than the supplied version number. |

| Syntax | is\_really\_writable(*$file*) |
| --- | --- |
| Parameters | $file (*string*) − File path |
| Return | TRUE if the path is writable, FALSE if not |
| Return Type | bool |
| Description | checks to see if file is writable or not. |

| Syntax | config\_item(*$key*) |
| --- | --- |
| Parameters | $key (*string*) − Config item key |
| Return | Configuration key value or NULL if not found |
| Return Type | mixed |
| Description | This function is used to get the configuration item |

| Syntax | set\_status\_header(*$code*[, *$text = ''*]) |
| --- | --- |
| Parameters | $code (*int*) − HTTP Response status code  $text (*string*) − A custom message to set with the status code |
| Return |  |
| Return Type | void |
| Description | This function permits you to manually set a server status header. |

| Syntax | remove\_invisible\_characters(*$str*[, *$url\_encoded = TRUE*]) |
| --- | --- |
| Parameters | $str (*string*) − Input string  $url\_encoded (*bool*) − Whether to remove URLencoded characters as well |
| Return | Sanitized string |
| Return Type | string |
| Description | This function prevents inserting NULL characters between ASCII characters |

| Syntax | html\_escape(*$var*) |
| --- | --- |
| Parameters | $var (*mixed*) − Variable to escape (string or array) |
| Return | HTML escaped string(s) |
| Return Type | mixed |
| Description | This function acts as a native PHP htmlspecialchars() function. |

| Syntax | get\_mimes() |
| --- | --- |
| Return | An associative array of file types |
| Return Type | array |
| Description | This function returns a reference to the MIMEs array from *application/config/mimes.php*. |

| Syntax | is\_https() |
| --- | --- |
| Return | TRUE if currently using HTTP-over-SSL, FALSE if not |
| Return Type | bool |
| Description | Returns TRUE if a secure (HTTPS) connection is used and FALSE in any other case (including non-HTTP requests). |

| Syntax | is\_cli() |
| --- | --- |
| Return | TRUE if currently running under CLI, FALSE otherwise |
| Return Type | bool |
| Description | Returns TRUE if the application is run through the command line and FALSE if not. |

| Syntax | function\_usable(*$function\_name*) |
| --- | --- |
| Parameters | $function\_name (*string*) − Function name |
| Return Type | bool |
| Description | Returns TRUE if a function exists and is usable, FALSE otherwise. |

118. How do you set default timezone in Codeigniter ?

-> PHP date\_default\_timezone\_set() function is used to sets the default timezone used by all date or time functions. In CodeIgniter, the best place to set the timezone is index. php file of application root. Place the date\_default\_timezone\_set() with your desired timezone in the main index.

119. What are the most prominent features of CodeIgniter?

-> A list of most prominent features of CodeIgniter:

* It is an open source framework and free to use.
* It is extremely light weighted.
* It is based on the Model View Controller (MVC) pattern.
* It has full featured database classes and support for several platforms.
* It is extensible. You can easily extend the system by using your libraries, helpers.
* Excellent documentation.

120. Explain the folder structure of CodeIgniter.

-> Application: The application folder is where all the code of the application we are developing is stored. It consists serval other folders.

* Cache: In this folder, all the cache pages of your application will be stored. This helps to increase the speed of your page access.
* Config: In this folder, all the configuration files are stored. Using database.php users can configure databases of applications and config.php files we set our base-url etc.
* Controllers: In this folder, it contains the control of our application and all server-side functionalities.
* Core: All the base classes of your application will be stored here.
* Helpers: This will help you in creating your application.
* Hooks: This will help you to modify the inner working of your framework.
* Language: You can use the language according to your need in the project.
* Logs: Here all the files related to the log will store and sometimes if you are getting errors and you are not getting the message then you can look into their explanation.
* Models: All the database logins will be stored here and the controller will request them here and then it can use them.
* Third\_party: All the third-party plugins will be stored here to use in the application.
* View: Here your all HTML files related to the application will be stored.

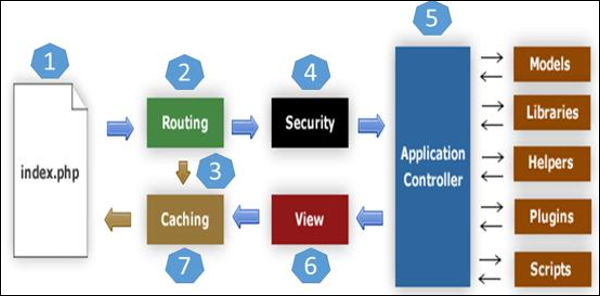
System: All the files related to coding, libraries, and other files will be stored here which will help you make the coding easy.

This folder also contains various folders which are explained below:

* Core: It consists of all the CodeIgniter’s core classes. Please do not try to make any change here.
* Database: All the drivers and utilities regarding the database will store here.
* Fonts: All the information and utilities regarding fonts are stored here.
* Helpers: It consists of all helpers-related data such as date, cookie etc.
* Languages: All language-related files stored here CodeIgniter supports multilingual web applications.
* Libraries: Here libraries will be stored which help you in creating applications easily like file upload, emails, calendars.

User\_guide: It works as an offline CodeIgniter guide which helps you to learn the basic functions of various libraries of CodeIgniter. You should go through this to learn about CodeIgniter. It consists of an index.php file

121. Explain CodeIgniter architecture.

-> 

* File index.php is the default file of CodeIgniter. It initializes the base resources.
* The Router decides what should be done with the information.
* If requested cache file exists, then the information is passed directly to the browser ignoring the further processes.
* Before loading Application Controller, the HTTP request and submitted data is passed under Security check.
* The Application Controller loads Models, Libraries, Helpers, Plugins and Scripts needed according to the request.
* The final page will come to View and then sent to the web browser. If View page is not cached then it will be cached first for future requests.
* Models – The database login is placed in this file.
* Helpers – The helper class of your application can be put in this folder.
* Libraries – The folder contains standard CodeIgniter libraries such as e-mail, calendars, file uploads etc. You can create your own libraries, extend and even replace standard ones. But those will be saved in the application/ libraries directory to keep them separate from the standard CodeIgniter libraries in this particular folder.

122. Explain model in CodeIgniter

-> In CodeIgniter Model are the PHP classes where all database related manipulation is done e.g. fetching records, insert, update, and delete records.

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

-> $this->load->model('ModelName');

if in case your model file is located in sub-directory of the model folder, then you have to mention the full path. For example, if your file location is application/controller/models/project/ModelName. Then, your file will be loaded as shown below,

$this->load->model('project/ModelName');

124. How can you connect models to a database manually?

-> $this->load->database();

125. Explain views in CodeIgniter.

-> views are special files used in CodeIgniter to store the markup outputted by the application, usually consisting of HTML and simple PHP tags. “A view is simply a web page, or a page fragment, like a header, footer, sidebar, etc. and design use

126. Explain controller in CodeIgniter.

-> A controller is the intermediary between models and views to process HTTP request and generates a web page. All the requests received by the controller are passed on to models and views to process the information. It is the center of every request on your web application.

127. What is the default controller in CodeIgniter?

-> The file specified in default controller will be loaded by default when no file name is mentioned in the URL. By default, it is Welcome.php which is the first page to be seen after installing CodeIgniter.

With URL

1. localhost/codeigniter/

Welcome.php will be loaded as there is no file name mentioned in the URL.

Although as per your need, you can change default controller in the file application/config/routes.php.

1. $route['default\_controller'] = ' ';

Here, specify your file name which you want to be loaded by default.

## Class Constructors

To use a constructor you need to mention the following line of code,

1. Parent::\_\_construct()

We need to manually call the parent constructor because local constructor will be overriding the one in the parent controller.

128. How will you call a constructor in CodeIgniter?

-> parent::\_construct()

129. What is an inhibitor of CodeIgniter?

-> In CodeIgniter, Inhibitor is an error handler class that uses native PHP functions like set\_exception\_handler, set\_error\_handler, register\_shutdown\_function to handle parse errors, exceptions, and fatal errors.

130. Explain the remapping method calls in CodeIgniter.

-> The Second segment of URI determines which method is being called. If you want to override it, you can use \_remap() method. The \_remap method always get called even if URI is different. It overrides the URI. For Example:

1. public function \_remap($methodName)
2. {
3. if ($methodName === 'a\_method')
4. {
5. $this->method();
6. }
7. else
8. {
9. $this->defaultMethod();
10. }
11. }

131. How can you load multiple helper files?

->

$this->load->helper(

1. array('helper1', 'helper2', 'helper3')
2. );