Q1. What is CodeIgniter?

Ans: CodeIgniter is a PHP MVC based Framework , which is used to develop website or web application rapidly by using their built in features. It is an Open Source framework. It has a very rich set of functionality, which will increase the speed of website development work.

What Is a Framework?

A framework is a structure that you can build software on. It serves as a foundation, so you're not starting entirely from scratch. Frameworks are typically associated with a **specific programming language** and are suited to different types of tasks.

You also have backend frameworks for backend tasks, frontend frameworks for frontend tasks, mobile frameworks for mobile tasks, and many others.

A software framework is a structure that you can use to build software. It acts as a foundation so you don't have to deal with creating unnecessary extra logic from scratch.

A framework is similar to a template in that you can modify it and add certain features and higher functionalities to create a complex and broad project that many people can use.

Why do we use frameworks?

- Using frameworks saves time and reduces the risk of errors. You don't need to write everything from the ground up, so there's less chance of introducing errors. Plus, frameworks have already been tested, so there's less to worry about. Other advantages include:
- More secure code
- Simpler testing and debugging
- Avoiding duplicate code
- Clean and easily adaptable code
- Able to focus on writing code specific to the project
- Can be extended

Libraries vs. frameworks

The terms "libraries" and "frameworks" are sometimes used interchangeably, but they're very different.

A framework is a supporting structure that requires specificity. You must follow the pattern of the framework.

A library is a set of previously-written code that you can use to build your own code.

We can say that a library implements a particular function. Some examples of popular libraries are **React**, and **JQuery**.

We can define a framework as a collection of libraries implementing a particular methodology. Some of the more commonly known frameworks are **Angular, Vue, Model View Controller**, and **Model View Presenter.**

Q2. Features of CodeIgniter?

Ans:

- Model-View-Controller Based System
- Extremely Light Weight
- Generate SEO friendly URLs
- Form and Data Validation
- Security and XSS Filtering
- Session Management
- Pagination
- Query Builder Database Support
- 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
- Support for Hooks and Class Extensions
- Large library of “helper” functions

Q3. What is MVC?

Ans:

The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry standard web development framework to create scalable and extensible projects.

Model: It contains Database related codes.

View: It contains Designing related codes.

Controller: Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

Q4. Use of autoload.php file in CodeIgniter?

Ans: autload.php file in CodeIgniter is used to load libraries, model , helper file. If we load model , libraries or helper in autoload.php file it is available for complete project there is no need to load model in each controller file.

Q5. Url Structure in CodeIgniter?

Ans: url structure in CodeIgniter is following:

Baseurl/index.php/controllername/functionname

Example:

http://localhost/php-classes/tts2/index.php/Welcome/about

Q6. How to remove index.php from URL in CodeIgniter?

Ans: using .htaccess file we can remove index.php from URL in CodeIgniter. Write the following code in .htaccess file:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L]
```

Q7. Routes in CodeIgniter?

Ans: routes.php file in CodeIgniter is used to create SEO Friendly urls.

Q8. Hooks in CodeIgniter?

Ans: Hooks in CodeIgniter is used for authentication, hooks like a middleware it checks the code before execute.

Q9. How to change the default Controller?

Ans: Default Controller of CodeIgniter is Welcome.php file, we can change the default controller from routes.php file

Example:

```
$route['default_controller'] = 'welcome';
```

Here place of welcome string change the controller name whatever you want.

Q10. What are helpers in CodeIgniter?

Ans:

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.

Q11. Difference between helper and library ?

Ans: The main difference between Helper and Library in CodeIgniter is that

Helper is a file with a set of functions in a particular category and is not written in Object Oriented format.

Library is a class with a set of functions that allows creating an instance of that class and is written in Object Oriented format.

Q12. How to create a custom library in CodeIgniter?

Ans: we can create custom library in a CodeIgniter in the libraries directory. When we create custom library , There are some points you should keep in mind:

Naming Conventions

- File names must be capitalized. For example: Myclass.php
- Class declarations must be capitalized. For example: class Myclass
- Class names and file names must match.

Example:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
 class Someclass {
public function some_method() { }
}
```

How to load library

```
$this->load->library('someclass');
```

Q13. Form Validation in CodeIgniter?

Ans: Form validation is crucial part of our web development , there are two types of form validation:

- **Client side validation:** In this we use JavaScript or jQuery to validate our form.
- **Server side validation:** In this we use PHP to validate our form.

To use Server Side validation , In a CodeIgniter there is a form validation library which is used to validate form easily, there is no need to write more line of code like in a core PHP.

Example:

```
$this->load->library('form_validation');

$this->form_validation->set_rules('name','Name','trim|required|min_length[4]|max_length[20]');
$this->form_validation->set_rules('email','Email','required|valid_email');
$this->form_validation->set_rules('mobile','phone','required|numeric|exact_length[10]');
```

Q14. Security in CodeIgniter?

Ans: Security methods in CodeIgniter are following:

XSS Prevention

XSS means cross-site scripting. CodeIgniter comes with XSS filtering security. This filter will prevent any malicious JavaScript code or any other code that attempts to hijack cookie and do malicious activities. To filter data through the XSS filter, use the **xss_clean()** method as shown below.

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

You should use this function only when you are submitting data. The optional second Boolean parameter can also be used to check image file for XSS attack. This is useful for file upload facility. If its value is true, means image is safe and not otherwise.

SQL Injection Prevention

- SQL injection is an attack made on database query. In PHP, we are use **mysql_real_escape_string()** function to prevent this along with other techniques but CodeIgniter provides inbuilt functions and libraries to prevent this.

We can prevent SQL Injection in CodeIgniter in the following three ways −

- Escaping Queries
- Query Biding
- Active Record Class

CSRF Prevention

- CSRF stands for cross-site request forgery. You can prevent this attack by enabling it in the **application/config/config.php** file as shown below.

$config['csrf_protection'] = TRUE;

- When you are creating form using **form_open()** function, it will automatically insert a CSRF as hidden field. You can also manually add the CSRF using the **get_csrf_token_name()** and **get_csrf_hash()** function. The **get_csrf_token_name()** function will return the name of the CSRF and **get_csrf_hash()** will return the hash value of CSRF.

- The CSRF token can be regenerated every time for submission or you can also keep it same throughout the life of CSRF cookie. By setting the value **TRUE**, in config array with key **'csrf_regenerate'** will regenerate token as shown below.

$config['csrf_regenerate'] = TRUE;