1. CI
2. Structure

* Application: source code of app
* config: contain all configuration files of application
* autoload.php: load libraries, helpers, models… automatically when a controller is initialized
* config.php: set base\_url,…
* routes.php: set default\_controller, re-map url,
* database.php: setting need to connect database
* cache: store the precessed data, support quick load, increase the speed of pages access
* **controllers: file control**
* **models: access with database**
* **views: store files UI to render**
* **core: contain file to custom core CI**

**prefix class: MY\_**

* helpers: function
* libraries: class
* language: support multi-languages
* third\_party: outside plugins
* logs: track application errors/exception
* hooks:
* System: core CI
* core:
* database:
* fonts:
* helpers:
* libraries:
* language:
* User\_guide: CI documentation

1. Usage:

* Route: [controller]/[action]/[params]
* Always extends CI\_Model/CI\_Controller for custom model/controller
* Use model in controller:

$this->load->model([name\_of\_model]);

$this->[name\_of\_model]->[action];

* Use library in controller:

$this->load-> library([name\_of\_library]);

$this->[name\_of\_library]->[action];

* Use helper in controller:

$this->load-> helper([name\_of\_helper]);

Function\_in\_helper();

* Load view in controller:

$this->load->view([name\_of\_ view], [data\_array]);

1. Integrate 3rd party

3 steps:

* Copy folder 3rd party library and parse into folder application/third\_party
* Create a library (in application/libraries) that is used to connect this 3rd party with CI

Include APPPATH.[path\_to\_3rd\_party\_library];

Class [library] extends [3rd\_party\_library] {...}

* Using 3rd party library like libraries in core CI

$this->load->library([name\_library]);

1. Smarty
2. Webapp
3. Web app
4. Web database
5. CDN
6. Header status code
7. Integrate HMVC CI – Smarty
8. Structure
9. Hoạt động
10. Demo
11. DB
12. InnoDB vs MyISAM
13. Fulltext search indexes