**CS-213 Advance Programming**

Assignment 1: Exploring Open Source Software

CLO 3: Develop applications and tools using various frameworks

**Topic: CAKE PHP**

**Submitted To**: Dr. Sidra Sultana

**Submitted by:** Zunaira Salam

**Registration No:** 214255

**Section:** B

Table of Contents

[**1.** Introduction: 3](#_Toc19561905)

[**2.** MVC Architecture: 3](#_Toc19561906)

[1.1. The Model Layer: 3](#_Toc19561907)

[1.2. The View Layer: 3](#_Toc19561908)

[1.3. The Controller Layer: 3](#_Toc19561909)

[**3.** The Request Cycle in CakePHP: 3](#_Toc19561910)

[**4.** Installation: 4](#_Toc19561911)

[**5.** Directories in CakePHP: 4](#_Toc19561912)

[**6.** Index Page: 5](#_Toc19561913)

[**7.** Styling: 6](#_Toc19561914)

[**8.** Database Connection: 6](#_Toc19561915)

[**9.** Views: 7](#_Toc19561916)

[**10.** AppController: 7](#_Toc19561917)

[**11.** Plugins: 7](#_Toc19561918)

# Introduction:

CakePHP is an open source rapid development framework for PHP. It provides a foundation structure for creating web applications. It has an organizational structure having classes, files, database tables and some conventions, which when followed makes a uniform application structure.

# MVC Architecture:

CakePHP is a framework built on MVC architecture. Model-View-Controller (MVC) is an architecture that divides an application into three main components: model, view and controller.

## The Model Layer:

The model layer of application implements business logic. This layer retrieves data and converts it to meaningful concepts in application. The data is processed, saved, associated, retrieved and validated in this layer.

## The View Layer:

This layer is responsible for presenting the modeled data. It produces presentational interface for the application using the available information. The View can render HTML template, XML, JSON and CSV from the model data.

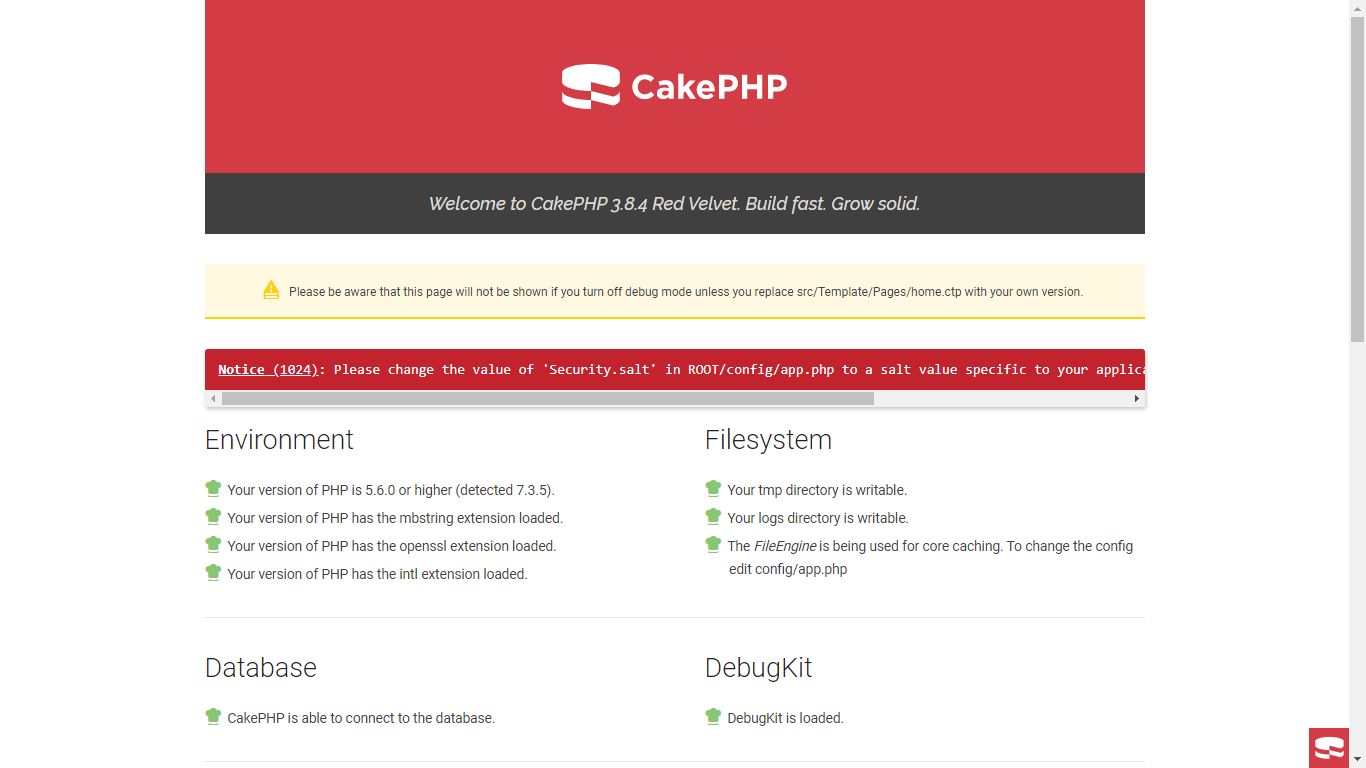
## The Controller Layer:

This layer is responsible for handling user requests. It renders a response with support of the Model and View layers. Controller manages the resources needed for completing a task and ensures their assignment to correct workers.

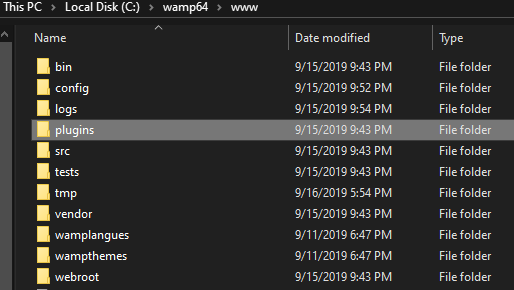
# The Request Cycle in CakePHP:

* + The webserver directs the request to **webroot/index.php**.
  + Application is loaded in an HttpServer.
  + Application’s middleware is initialized.
  + Controllers and actions are selected.
  + The controller’s action is called, and the controller interacts with the required Models and Components.
  + The controller delegates response creation to the View to generate the output resulting from the model data.
  + The view uses Helpers and Cells to generate the response body and headers.
  + The response is sent back out through the [Middleware](https://book.cakephp.org/3.0/en/controllers/middleware.html).
  + The HttpServer emits the response to the webserver.

# Installation:

Download the zip file for CakePHP from their website or github. Install an apache server on your system like WAMP or XAMP etc. Extract the CakePHP zip file in the **www** folder in WAMP server. Go to Wamp/www/config/app.default.php and rename the file to app.php. Now run the local host and the index page of CakePhp is displayed.

# Directories in CakePHP:

1. Bin folder: the command line executable files for cake are present in this folder.
2. Config folder: the configuration files, database configuration details and core configuration files are present in this folder.
3. Plugins folder: the cakePHP plugins for applications are stored in this folder.
4. Src folder: It contains the source files for the cake application. It contains the following sub folders:
   * Console
   * Controller
   * Middleware
   * Model
   * Shell
   * Template
   * View
5. Tests: this folder contains the test cases for cake application.
6. Vendor: this folder contains CakePHP and application dependencies. It is not recommended to change the contents of this folder.
7. Webroot: this is the public directory for the application where all the publicly reachable files are present.

# Index Page:

The CakePHP index page can be modifies by editing the file **src/Template/Pages/home.ctp.** This file contains HTML and php for basic functionality like database connection, filesystem checks and php extension checks.

# Styling:

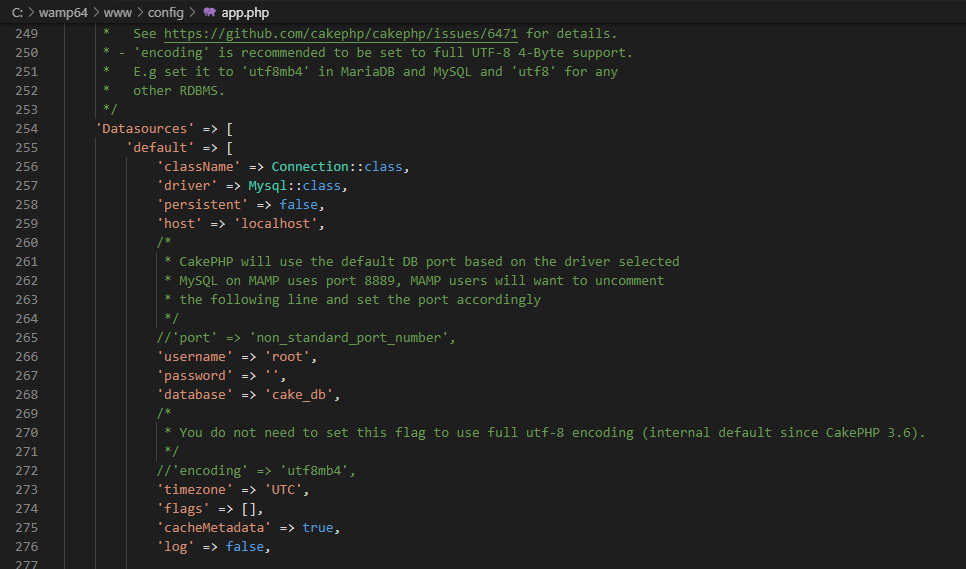
We can change the CSS of index page or any other in the **webroot/css/** folder and styles to the application. The fonts and images should also be present in this folder.

# Database Connection:

At first the index page will show an error in connecting with database. First create a database in MySql in the WAMP server and create tables as required. Go to **config** folder in www and open the file **app.php**. it contains various database variables which can be modified as per user requirements. Some of the variables are:

* ClassName which represents the connection to the database server.
* Driver represents the class name of the driver which has been used to implement the database.
* Database server’s host name
* Username of database
* Password of the database
* Name of the database

Set the database connection by editing the host name, password and database name. save the file and refresh the local host. Now the database will be connected.



CakePHP has a lot of built in libraries for creating, reading, updating and deleting data in database. These libraries provide methods like hashing to store passwords, a table manager and database connection manager.

# Views:

The View classes send an output to user in response to the request. These View classes are very powerful and can speed up the development process. The view template files are present in the **Template** folder. These files have an extension**.ctp (CakePHP template)**. Templates are responsible for getting data from controller, renderit and send and output.

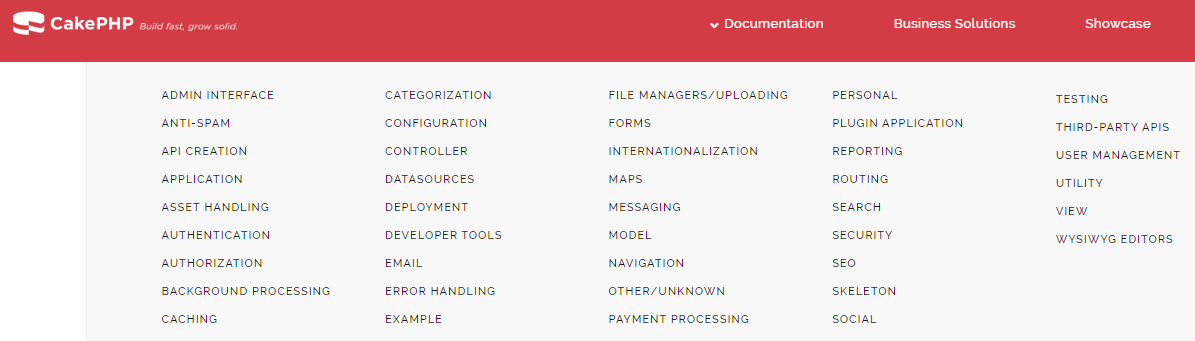
# AppController:

The parent class of all application’s controllers is AppController. This class is defined in the src/controller/AppController.php. This class is used to initialize or load components using the built-in **initialize()** method in the constructor. **Actions** are the methods in the controller class. These actions handle the responses for browser requests.

# Plugins:

Themes or plugins in CakePHP can be used to add styles, switch themes or add small functionalities in applications. We can create our own plugins as well. The files of theme template should reside in a plugin with the same name as the theme. The theme name should be in CamelCase.

The modules present for CakePHP are following:



AdminLTE is a CakePHP based plugin to develop business dashboard.

