MINDFULNESS AT MONASH

Technical Documentation

ABSTRACT
This document serves to educate the handover team on who the appropriate personnel are, how they will maintain the system as well as make any appropriate changes to the database.

Gideon Jian Yi Swee
FIT3048

Contents

Introduction

The technical documentation is for personnel working with the source code of Mindfulness at Monash's website in cases where tasks beyond users' expertise, such as creating new functionality or troubleshooting errors, need to be done.

Understanding its implementation of the CakePHP framework and the correct syntax is crucial to efficient and correct implementation.

Audience

This documentation is meant for the technical support staff of Mindfulness at Monash. These include the manager, administrators and programmers.

Readers of this documentation should have experience in PHP programming, preferably in the CakePHP framework. Experience with version control tools like Git is not necessary, but beneficial for understanding how to install, update and deploy this project.

For those without experience, there is a very basic guide to version control in the Deployment section.

Purpose

The Mindfulness at Monash CMS source code is built to handle most of the technical heavy lifting of:

- formatting plain text into professional-looking articles,
- displaying data from the database as front end info,
- making changes to data without manually accessing the database, and
- creating new admins & deleting old ones.

This documentation concerns understanding how CakePHP's MVC framework implements the website.

Scope

This documentation covers the underlying source code of the Mindfulness at Monash website. Both the front end website and back end administrator operations will be covered.

At times, the CakePHP documentation will be referenced when certain built-in functions are used.

Application Specifications

Frameworks & Languages

- **CakePHP 3.6**
  - Bootstrap 4 (JavaScript & CSS framework)
  - .CTP (CakePHP's proprietary version of HTML/PHP)
  - .CSS
  - .JS (JavaScript)
- **PHP 7.2**
- **MySQL**
- **jQuery**

Templates

- Clean Blog (https://startbootstrap.com/template-overviews/clean-blog/)

Third-party Modules

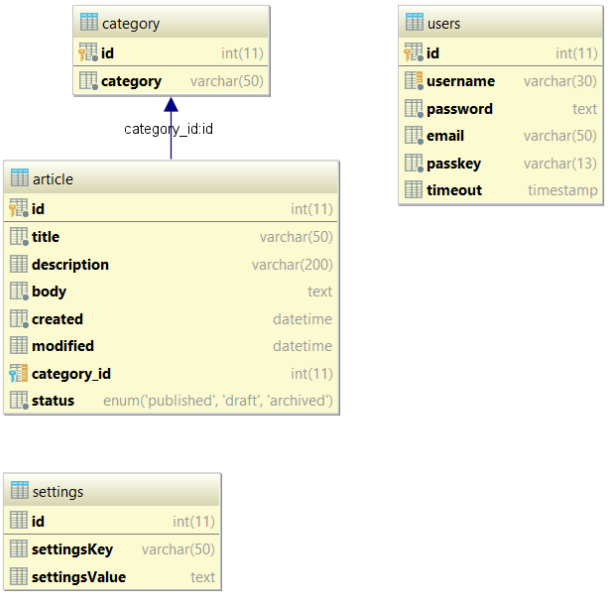
- TinyMCE 5.0
- Gmail SMTP server

Database Information

Entity Relationship Diagram

Each article has a category – in table **article** each article's (row) category is identified with FK category_id, which gives the category name in table **category**.

Tables **users** & **settings** are separate from the article table. Users is purely for authentication and settings purely for editing home page information.



Restore from Backup

The entire Mindfulness at Monash web application is stored on a Git server: https://git.infotech.monash.edu/UGIE/UGIE-2018/team121.

This allows multiple devices to make changes to the same project without having to exchange data among themselves, making team effort more efficient. It also reduces the risk of losing work from local hardware failure and increases ease of data migration, as all data is stored on the cloud.

To transfer the files over to your computer, you'll need to learn how to restore it from the Git server.

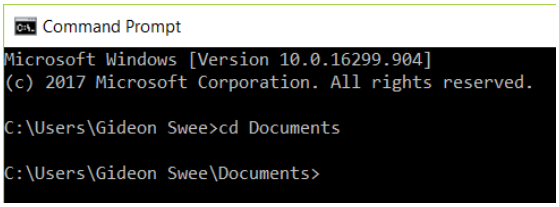
Prerequisites:

- Git (Download for Mac, Windows or Linux here: https://git-scm.com/downloads)

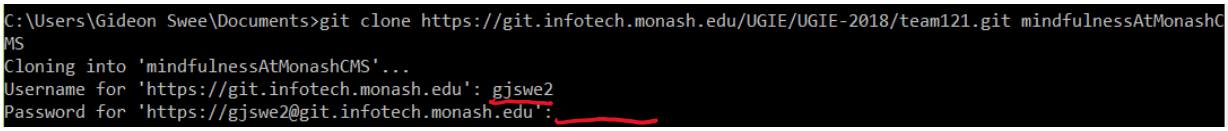
Cloning

To install the Git repository on your computer:

1. Open Command Prompt & navigate to the location of your folder using `cd`.
   e.g. Documents

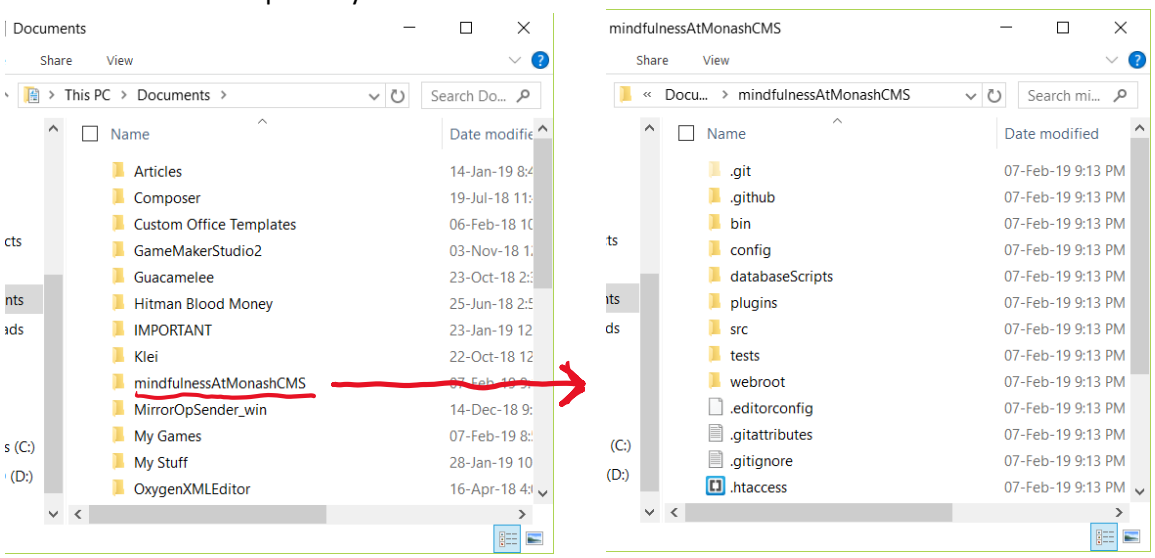


2. Enter `git clone https://git.infotech.monash.edu/UGIE/UGIE-2018/team121.git mindfulnessAtMonashCMS`



   Enter the username and then enter the password (will not show on screen) in the popup:
   Username: `gjswe2`
   Password: `gideonswee`

3. You will see the Git repository installed as folder ‘mindfulnessAtMonashCMS’.



Setting Up

Due to security reasons, **‘autoload’ files** & **app.php** are not included within the folder’s files. These specific files are required for running any CakePHP application.

Autoload files

If you run the command `cake server` within the **‘bin’** folder of the project, a bunch of errors appear which means **the ‘autoload’ files are missing**.

```
C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\bin>cake server
PHP Warning:  require(C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS/vendor/autoload.php): failed to open stream:
 No such file or directory in C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\bin\cake.php on line 5

Warning: require(C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS/vendor/autoload.php): failed to open stream: No s
uch file or directory in C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\bin\cake.php on line 5
PHP Fatal error:  require(): Failed opening required 'C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS/vendor/autol
oad.php' (include_path='C:\xampp\php\PEAR') in C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\bin\cake.php on lin
e 5

Fatal error: require(): Failed opening required 'C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS/vendor/autoload.p
hp' (include_path='C:\xampp\php\PEAR') in C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\bin\cake.php on line 5


C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\bin>
```

We install these ‘autoload’ files through a tool called Composer.

If you already have Composer installed on your computer, skip to the next page. If not, follow the instructions below to install Composer.

There are two methods of installing Composer: for Windows & non-Windows OS’s. If the project is on a Windows computer:

1. **Download the Composer global installer** from the website: https://getcomposer.org/download/
2. Install Composer
3. Run command `composer install` in the root folder.

If the project is not on Windows, refer to the section "**Command-line installation**" on the Composer website.

To install the 'autoload' files, enter composer install in the root folder and wait for the vendor & 'autoload' files to install.

```
C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS>composer install
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Package operations: 46 installs, 0 updates, 0 removals
  - Installing cakephp/plugin-installer (1.1.0): Loading from cache
  - Installing aura/intl (3.0.0): Loading from cache
  - Installing symfony/yaml (v3.4.4): Loading from cache
  - Installing symfony/polyfill-mbstring (v1.7.0): Loading from cache
```

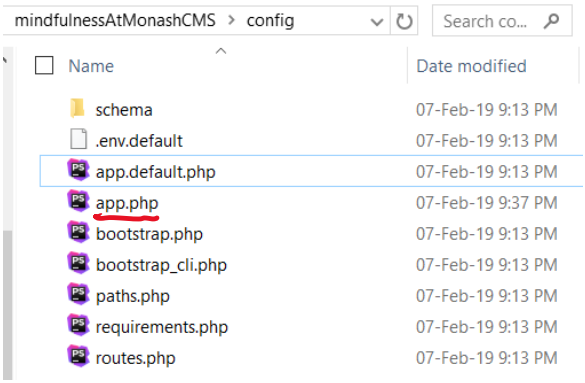
...

```
Set Folder Permissions ? (Default to Y) [Y,n]? y
Updated Security.salt value in config/app.php

C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS>
```

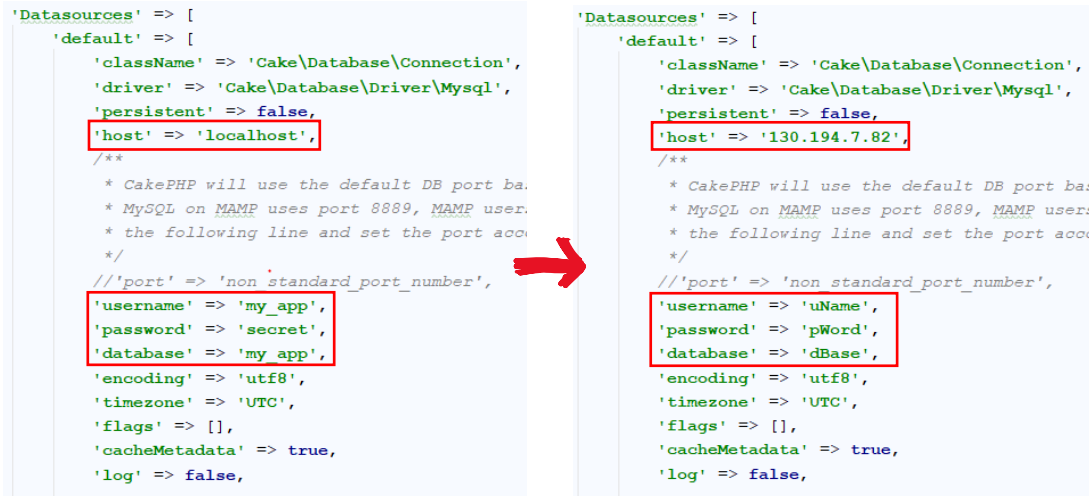
At the end, enter 'Y'.

app.php

After running composer install you will find the app.php file installed in the config folder.



Open the file. Inside, the boxed fields below should be filled in with the *connection details* of your Database. For example (actual connection details will be provided in private):



We need to connect to a database, otherwise when accessing any page that retrieves data from the database, the following error will appear:

Running the Application

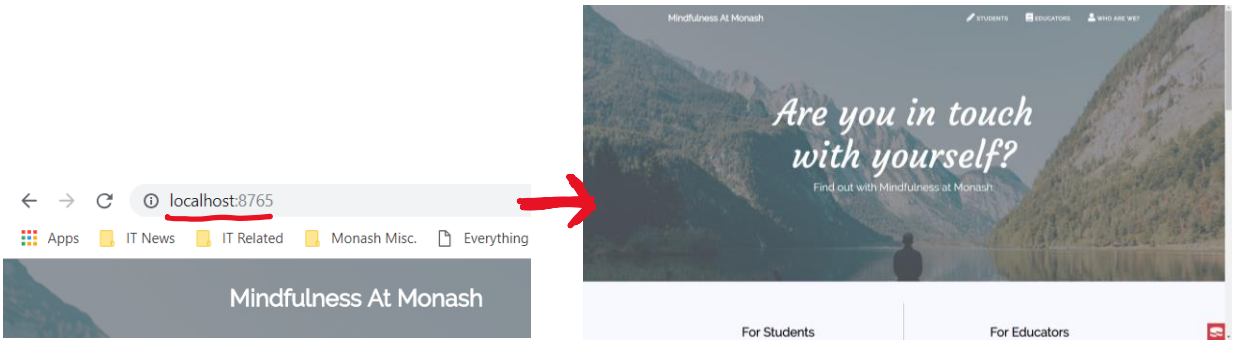
To run, open Command Prompt and navigate to the bin section of your project.

```
C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS>cd bin

C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\bin>
```

Enter `cake server` and the following should appear:

```
C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\bin>cake server

Welcome to CakePHP v3.5.12 Console
---------------------------------------------------------------
App : src
Path: C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\src\
DocumentRoot: C:\Users\Gideon Swee\Documents\mindfulnessAtMonashCMS\webroot
Ini Path:
---------------------------------------------------------------
built-in server is running in http://localhost:8765/
You can exit with `CTRL-C`
```

Open your web browser and enter the URL `localhost:8765`. If working, the front page should appear:



If everything works, we can move on to how this project implements CakePHP.

Implementation

CakePHP is a web application framework that creates a web application with 2 ends – the Client frontend & Admin backend.

- The Client frontend displays data from a connected database (for this project, the database on page 2).
- The Admin backend manages data in that database.

Both similarly rely on different folders of **source files** to display content on screen.

This section concerns how this project makes changes to the default system to provide project-specific functionalities and UI changes.

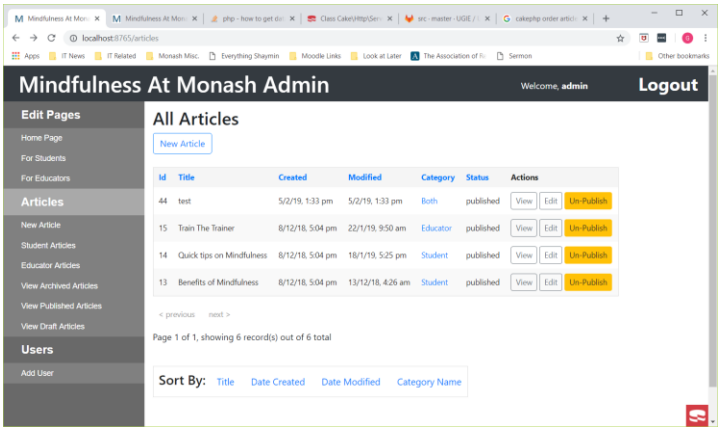
Layout

The function `$this->viewBuilder()->setLayout('nameOfLayout')` can be declared in the `initialize()` function for any Controller to set it as the default layout (e.g. ArticleController):

```
public function initialize()
{
    parent::initialize();
    //$this->loadModel('User');
    $this->Auth->allow(['view','viewArticleIndex']);
    $this->viewBuilder()->setLayout('admin');
}
```

(don't worry about Auth – it'll be explained in Security later)

This gives us the layout:

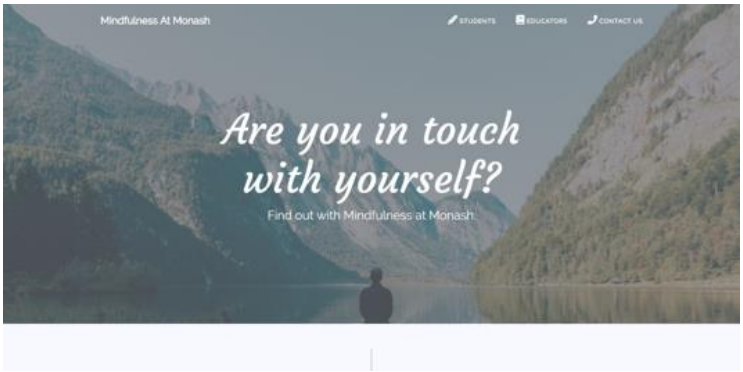


But it can also be declared inside a function to make the representative page inherit that layout (e.g. ArticleController):

```
public function view($id = null)
{
    $article = $this->Article->get($id, [
        'contain' => ['Category']

    ]);
    // If the article is a draft or archived - throw exception
    if ($article->get('status') == 'draft' or $article->get('status') ==
```

```
'archived')
    {
        if (is_null($this->request->session()->read('Auth.User.username'))) {
            throw new NotFoundException(__('Article not found'));
        } else {
            $this->set('article', $article);
            $this->viewBuilder()->setLayout('default');
        }

    } else {
        $this->set('article', $article);
        $this->viewBuilder()->setLayout('default');
    }

}
```

This means that the corresponding page, view.ctp, will be displayed with the 'default.ctp' layout instead of the 'admin.ctp' layout, giving us the layout:



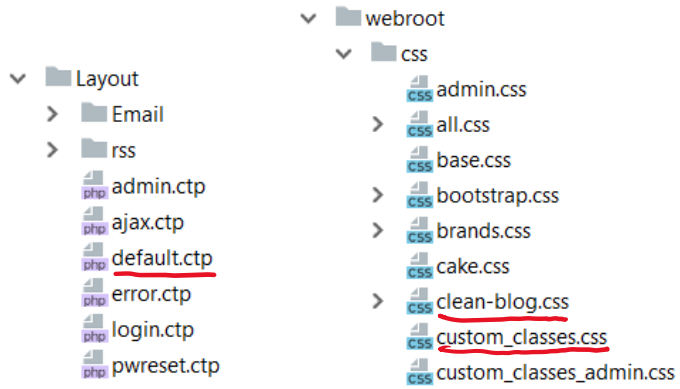
Regardless of layout type, all layout .ctp files must be stored in the Layout folder.

There are three layouts: the Website, Admin & Login layouts.

Listed below are source files that use each layout (and in brackets, the corresponding Controller file), with the files that were edited to obtain it.

Website

Files to edit:

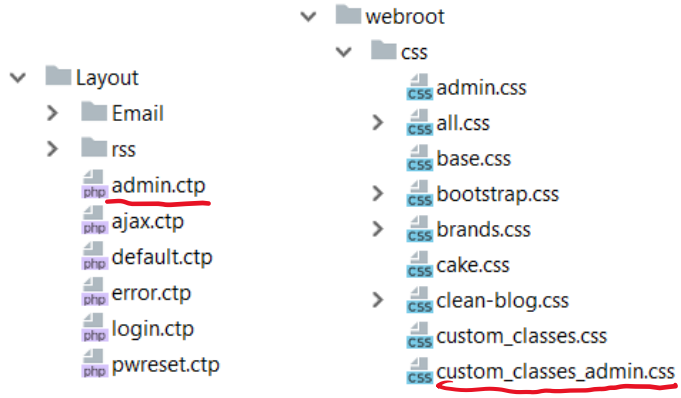


Files using the layout:

- home.ctp
- **ArticlesController**
    - view.ctp
    - viewArticleIndex.ctp

Admin

Files edited:

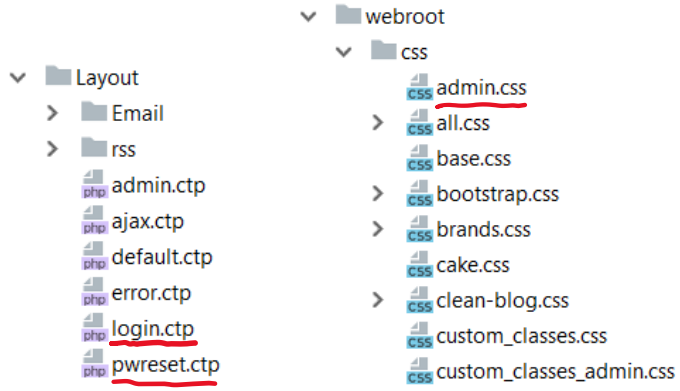


Files using the layout:

- **ArticlesController**
    - add.ctp
    - edit.ctp
    - index.ctp
    - searchByCategory.ctp
    - searchByStatus.ctp
- **UsersController**
    - add.ctp
    - admin_home.ctp
    - edit.ctp
    - index.ctp
    - view.ctp
- **SettingsController**
    - edit.ctp
    - index.ctp

Login

Files edited:



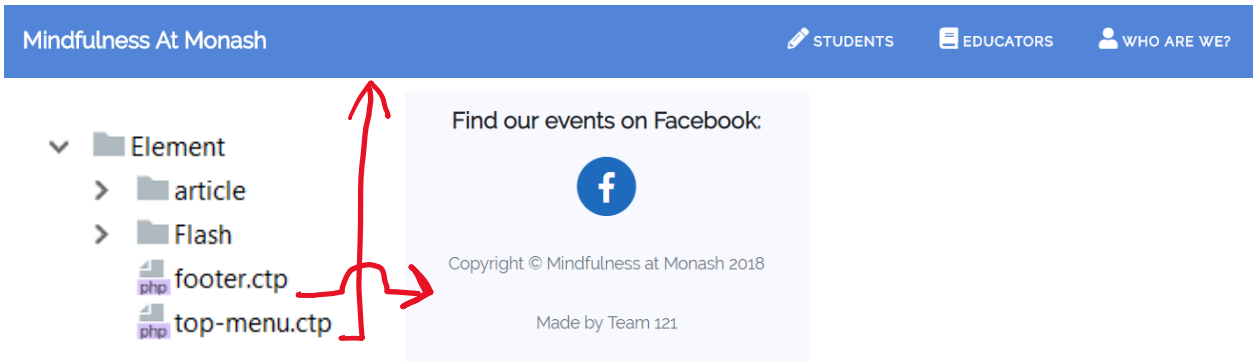
Files using the layout:

- **UsersController**
  - login.ctp
  - password.ctp
  - reset.ctp

Element

Elements are common parts of the website used across source & layout files.

In default.ctp, the top bar & footer are represented by the files top-menu.ctp & footer.ctp.



Ignore folders 'article' & 'Flash'. Files in 'article' were not used, while files in 'Flash' should not be edited.

Webroot

Folders with css, js libraries, img files and files of other formats are stored here. Cake references them using the HTML helper:

```
<?= $this->Html->script('jquery-3.2.1.min.js') ?>
```

JavaScript

```
<?= $this->Html->css('admin.css') ?>
```

CSS

In this web application, all images are stored in Imgur to reduce issues of server migration. Each image is referenced with a direct link.

The Imgur account can be found here: https://mindfulnessatmonash.imgur.com/
Password: `gidgid`

Other formats can be found in the documentation:
https://book.cakephp.org/3.0/en/views/helpers/html.html

Security

The login system was implemented according to the CakePHP 3 Cookbook:
https://book.cakephp.org/3.0/en/controllers/components/authentication.html

To make pages accessible to the public, in any Controller file, use the function `$this->Auth->allow` to make pages public. Include functions **within** the Controller.

E.g. ArticleController:

```
public function initialize()
{
    parent::initialize();
    //$this->loadModel('User');
    $this->Auth->allow(['view','viewArticleIndex']);
    $this->viewBuilder()->setLayout('admin');
}
```

This makes the pages view.ctp & viewArticleIndex.ctp publicly accessible.

Third Party Modules

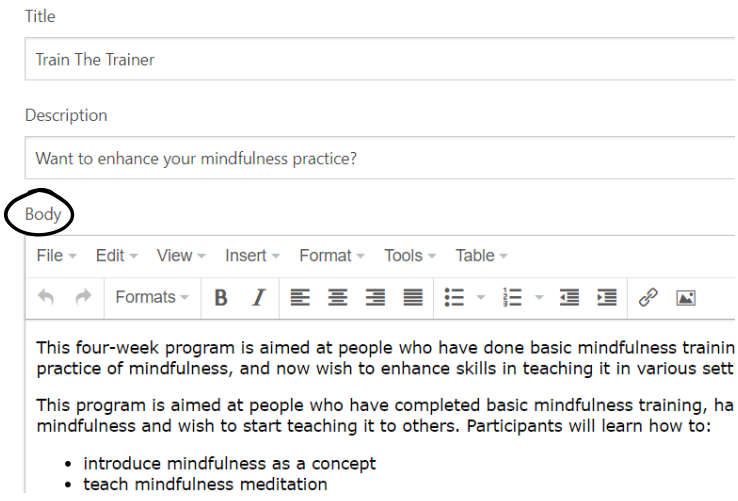
TinyMCE 5.0

TinyMCE is self-hosted. The text editor is declared in src\Template\Layout\admin.ctp:

```
<!DOCTYPE html>
<html>
<head>
    <?= $this->Html->script("tinymce/tinymce.min.js") ?>
    <script type="text/javascript">
        tinymce.init({
        selector: '#body',
          plugins: ["advlist autolink lists link image charmap print preview anchor",
        "searchreplace visualblocks code fullscreen",
        "insertdatetime media table contextmenu paste autoresize"],
          //invalid_elements: 'br'
          toolbar: "insertfile undo redo | styleselect | bold italic | alignleft
           aligncenter alignright alignjustify | bullist numlist outdent indent |
           link image",
            file_picker_types: 'file image media',
              force_br_newlines : false,
              force_p_newlines : false,
            forced_root_block: '',
            invalid_elements:"div"
      });
    </script>
```

Text boxes with the fieldname (circled below) included in the selector (circled above) are converted to TinyMCE input boxes.

```
echo $this->Form->control( fieldName: 'title');
echo $this->Form->control( fieldName: 'description');

echo $this->Form->control( fieldName: 'body');
```



Documentation about plugins can be found in: https://www.tiny.cloud/docs/

Gmail SMTP

This web application uses Gmail for the SMTP in the case of an admin having to reset the password. Using Gmail is easier and more secure than setting up a personal SMTP server. Enforced TLS (Transport Layer Security) is used for security. The default email address used is mindfulnessatmonash@gmail.com.

If you are switching mail providers, make sure that the application is configured appropriately. Settings can be found here: https://support.google.com/a/answer/176600?hl=en

Information about Gmail's SMTP settings can be found here: https://support.google.com/a/answer/176600?hl=en

1. Go to the application's app.php file.
2. Edit the details in the app.php file (highlighted by the boxes):

The SMTP configuration details are written in the app.php file. Edit the default app.php file to use Gmail SMTP:

```
'EmailTransport' => [
    'default' => [
        'className' => 'Mail',
        // The following keys are used in SMTP transports
        'host' => 'localhost',
        'port' => 25,
        'timeout' => 30,
        'username' => null,
        'password' => null,
        'client' => null,
        'tls' => null,
        'url' => env('EMAIL_TRANSPORT_DEFAULT_URL', null),
    ],
],
```

```
'EmailTransport' => [
    'default' => [
        'className' => 'Smtp',
        // The following keys are used in SMTP transports
        'host' => 'tls://smtp.gmail.com',
        'port' => 465,
        'timeout' => 30,
        'username' => 'mindfulnessatmonash@gmail.com',
        'password' => 'm1ndfulness@monash',
        'client' => null,
        'tls' => false,
        'url' => env('EMAIL_TRANSPORT_DEFAULT_URL', null),
        'context' => [ //This section disables server verification
            'ssl' => [ //to avoid no ca-cert issue
                'verify_peer' => false,
                'verify_peer_name' => false,
                'allow_self_signed' => true
            ],
```

← additional

```
            ],
        ],
],
```

Change Datetime Format

In app.php, change the second parameter in ‘defaultLocale’ from ‘en-US’ to ‘en-AU’.

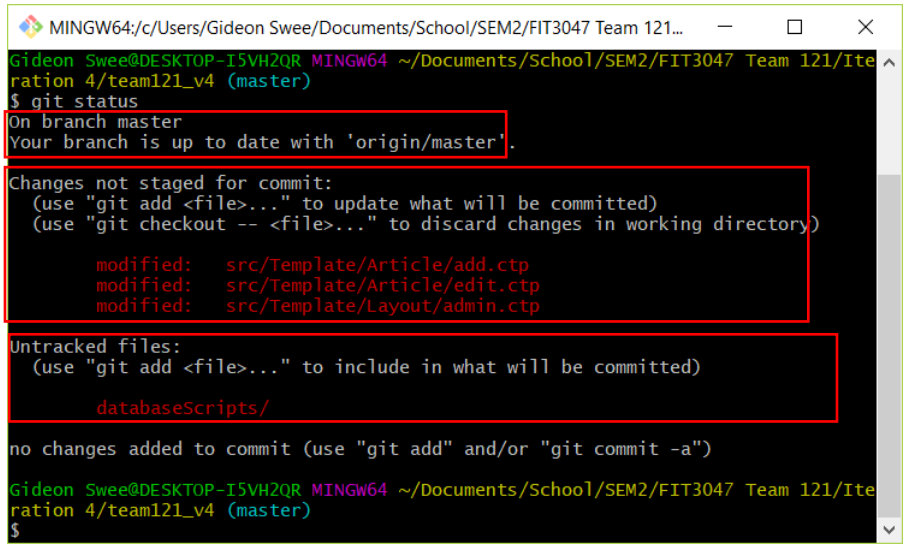
Deployment

This project must be committed to a Git repository before it can be 'pulled' (or updated) into the client's remote server.

If the project already has a Git repository, skip to Committing. Otherwise, follow the steps in https://kbroman.org/github_tutorial/pages/init.html to initialise your project in a Git repository.

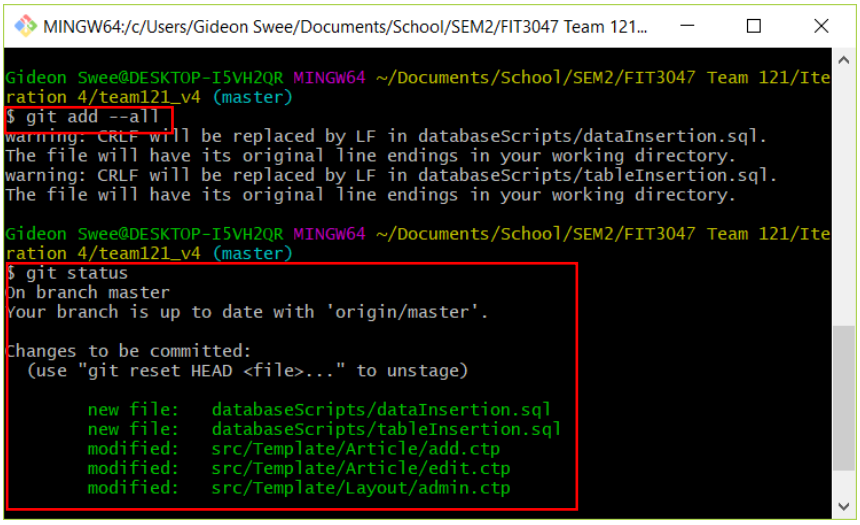
Committing

1. To see all changed or new ("untracked") files from the cloned repository (shown here as branch "master"), enter "git status"



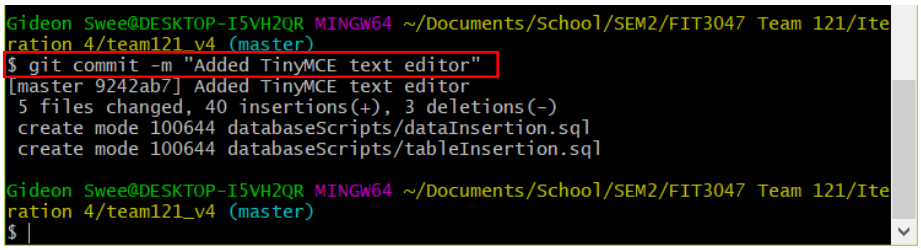
1. "Changes not staged…" shows files that have been changed from the repository online, but have not been included in the commit
2. "Untracked files" shows files that are new to the repository
3. "No changes added to commit" means no files have been included for the commit yet.

2. You can enter `git add` [file name e.g. `edit.ctp`] to add a specific file. But for convenience's sake, we'll add all files.
   To add all files to the commit, enter `git add --all`

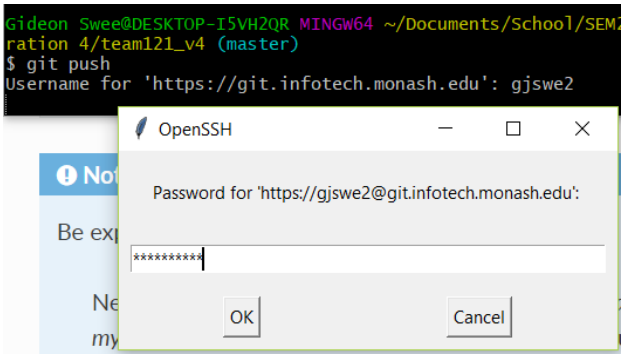


`git add --all` will add all uncommitted files into the next commit. Git status will show which files are new and which are modified from the repository.

3. Enter `git commit` followed by `-m "`[enter summary of changes here e.g. `Added text editor`]"



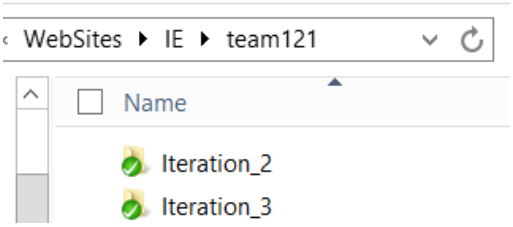
4. Finally – enter `git push` to push the commit to the Git repository. (if you are pushing to a branch, be explicit and state the repository & branch.) Enter the username and password of that Git repository.



5. Check your git repository whether the commit was made.

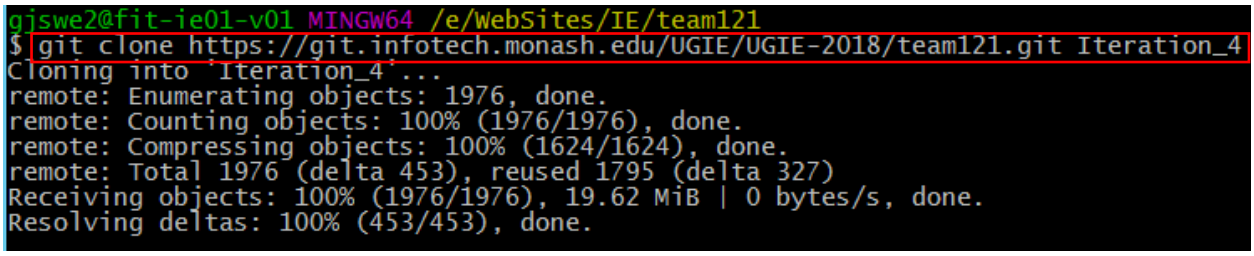
Remote Server Deployment

Let's say you have a 'main folder' where all your different websites are stored.

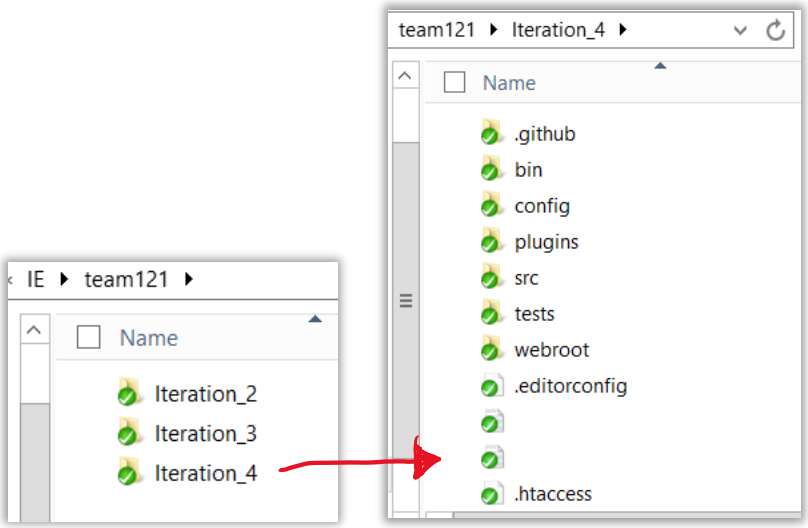


Preparing to Commit

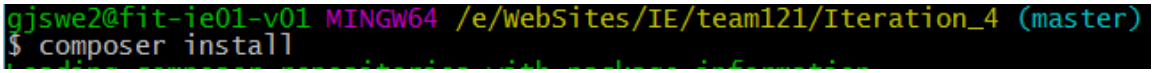
1. Run command prompt in the main folder and enter the code inside the red box:



This will create a new folder called "Iteration_4" which has the same exact files as the Git repository. You can also commit changes made in Iteration_4 to the repository from the remote server.



2. In the root of the new folder, enter "composer install" to load vendor & autoload files.



You'll have to wait until they ask "Set Folder Permissions? [y/n]". Select Y (though it

automatically sets Y by default).

```
Generating autoload files
> Cake\Composer\Installer\PluginInstaller::postAutoloadDump
> App\Console\Installer::postInstall
Created `config/app.php` file
Created `E:\WebSites\IE\team121\Iteration_4/logs` directory
Created `E:\WebSites\IE\team121\Iteration_4/tmp` directory
Created `E:\WebSites\IE\team121\Iteration_4/tmp/cache` directory
Created `E:\WebSites\IE\team121\Iteration_4/tmp/cache/models` directory
Created `E:\WebSites\IE\team121\Iteration_4/tmp/cache/persistent` directory
Created `E:\WebSites\IE\team121\Iteration_4/tmp/cache/views` directory
Created `E:\WebSites\IE\team121\Iteration_4/tmp/sessions` directory
Created `E:\WebSites\IE\team121\Iteration_4/tmp/tests` directory
Set Folder Permissions ? (Default to Y) [Y,n]? y
Updated Security.salt value in config/app.php
```

3. composer install will also install the app.php file in your config folder.

```
'Datasources' => [
    'default' => [
        'className' =>
        'Cake\Database\Connection',
        'driver' =>
        'Cake\Database\Driver\Mysql',
        'persistent' => false,
        'host' => 'localhost',
        /* ... */
        //'port' =>
        'non_standard_port_number',
        'username' => '',
        'password' => '',
        'database' => '',
```

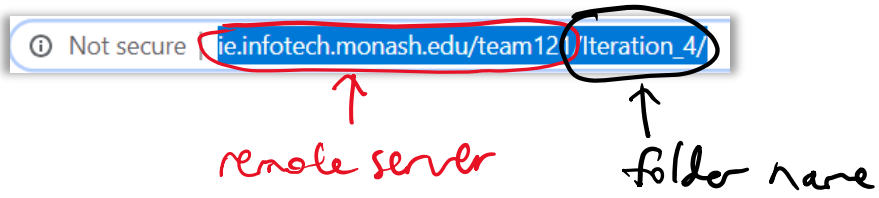
Fill the **circled** fields in the Datasources module with your host address, username, password and database name respectively.

In addition, fill the **circled** fields in the EmailTransport module with the type of transport protocol (className), host, port, username, password & context. Additional info is in red.

```
'EmailTransport' => [
    'default' => [
        'className' => 'Smtp',
        // The following keys are used in SMTP transports
        'host' => 'tls://smtp.gmail.com',
        'port' => 465,
        'timeout' => 30,
        'username' => '[redacted]@gmail.com',
        'password' => '[redacted]',
        'client' => null,
        'tls' => false,
        'url' => env( key: 'EMAIL_TRANSPORT_DEFAULT_URL', default: null),
        'context' => [ //This section disables server verification
            'ssl' => [ //to avoid no ca-cert issue
                'verify_peer' => false,
                'verify_peer_name' => false,
                'allow_self_signed' => true
            ],
        ],
    ],
],
```

This web application is based in Australia, so change the second parameter of 'defaultLocale' from 'en-US' to 'en-AU'.

4. To access your deployed website, in your browser address bar, enter the url of your remote server, followed by the folder name.



Database Creation

Create your database in your server's MySQL database manager e.g. PHPMyAdmin.

First, simply copy the code from databaseScripts\tableInsertion.sql into an SQL console and execute.

Next, do the same for databaseScripts\dataInsertion.sql.